DAA/ames

LINEAR AND NONLINEAR DYNAMIC ANALYSIS OF

REDUNDANT LOAD PATH BEARINGLESS ROTOR SYSTEMS

by

V.R. Murthy, Principal Investigator

Status Report

NASA, Ames Research Center

under

NASA Research Grant NAG2-306

December 1985

LINEAR AND NONLINEAR DYNAMIC ANALYSIS OF

REDUNDANT LOAD PATH BEARINGLESS ROTOR SYSTEMS

9813762

CS

by

V.R. Murthy, Principal Investigator

Associate Professor
Department of Mechanical & Aerospace Engineering
Syracuse University
Syracuse, NY 13244
Tel: (315) 423-2022

Status Report

Prepared for

NASA, Ames Research Center
University Affairs Office, M.S. 241-25
Moffett Field, CA 94035

under

NASA Research Grant NAG2-306

ATTN: Randall L. Peterson
Technical Officer
Low Speed Aircraft Branch
M.S. 247-1
Tel: (415) 694-5044

December 1985

Table of Contents

ABSTRACT

The bearingless rotorcraft offers reduced weight, less complexity and superior flying qualities. The bearingless rotors are presently being developed and it is most likely that the next generation rotorcraft would be equipped with these rotors. Almost all practical designs of bearingless rotors include multiple load paths and the one that was flight tested by the Boeing Vertol has three load paths. The determination of natural vibration characteristics is basic to any dynamic design and they form the basis for all practical aeroelastic stability analyses of rotor blades. Almost all the current industrial structural dynamic programs of conventional rotors which consist of single load path rotor blades employ the transfer matrix method because this method is ideally suited for one-dimensional chain-like structures. In this report, this method is extended to multiple load path rotor blades without resorting to an equivalent single load path approximation. Unlike the conventional blades, it is necessary to introduce the axial-degree-of-freedom into the solution process to account for the differential axial displacements in the different load paths. With the present extension, the current rotor dynamic programs can be modified with relative ease to account for the multiple load paths without resorting to the equivalent single load path modeling. The results obtained by the transfer matrix method are validated by comparing with the finite-element solutions. A differential stiffness matrix due to blade rotation is derived to facilitate the finite-element solutions.

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A	Area of cross-section
B_1^*, B_2^*	Cross-section integrals, reference 8
C_1, C_1^*	Cross-section integrals, reference 8
e	Mass centroid offset from elastic axis, positive when in front of elastic axis
e_A	Distance between area centroid and elastic axis, positive for centroid forward
E	Young's modulus
G	Shear modulus
I_y	Cross-section moment of inertia about y-axis
$I_{y'_o}$	Reference moment of inertia for non-dimensionalization
I_z	Cross-section moment of inertia about z-axis
J	Torsional rigidity
k_A	Polar radius of gyration of cross-sectional area effective in carrying tensile stresses about elastic axis
k_m	Mass radius of gyration of blade cross section $k_m^2 = k_{m_1}^2 + k_{m_2}^2$
k_{m_1}, k_{m_2}	Cross-section integrals, reference 8
ℓ	Length of the load paths
m	Mass per unit length
m_o	Reference mass for non-dimensionalization
M_x	Twisting moment about x-axis
M_y, M_z	Bending moments about y and z axes, respectively
n	Number of load paths
N	Axial force
R	Radius of the rotor
T	Tension

$[T]$	Transfer matrix of the blade
$[T^i]$	Transfer matrix of the ith load path
u, v, w	Elastic displacements in the x, y, z directions, respectively
V_y, V_z	Shear forces along y and z directions, respectively
x, y, z	Mutually perpendicular axis system with x along the undeformed blade and y towards the leading edge
$\{z\}$	State vector
ψ	Slope of deflection curve normal to plane of rotation
ν	Slope of deflection curve in the plane of rotation
θ	Pretwist angle
ϕ	Elastic twist about the elastic axis
ω	Frequency of vibration
Ω	Blade rotational speed
β_{pc}	Precone angle

Superscripts

'	Differentiation with respect to x
.	Differentiation with respect to time
-	Non-dimensional quantity
i	Quantities corresponding to the ith load path

Subscripts

1	Quantities at the root
2	Quantities at the clevis
3	Quantities at the blade tip
i	Quantities corresponding to the ith load path

I. INTRODUCTION

The bearingless rotorcraft offers simplicity of the design and superior flying qualities. The original purpose of introducing hinges is to relieve the blades from high stresses and this is an important design problem for bearingless rotors and a solution to this problem lies in the optimization of blade root stiffness distribution and the use of advanced composite materials. The bearingless rotor technology was successfully applied by Sikorsky in Blackhawk and S-76 helicopters for their tail rotors (Ref. 1). Boeing Vertol built the first successful bearingless main rotor (Refs. 2 and 3) and this flew first in 1978. During a recent study (Ref. 4) for concept definition of the Integrated Technology Rotor/Flight Research Rotor (ITR/FRR), thirty-three hub concepts were proposed. Twenty-one out of these thirty-three concepts were bearingless designs and hence it is very likely that the next generation rotorcraft would be equipped with a bearingless rotor.

Basically, the structural design of bearingless rotors include multiple load paths and also some kinematic couplings are introduced intentionally through design for specific reasons. It is evident from Refs. 4 to 7 that several flexbeam and pitch control configurations are possible for bearingless main rotors and the variations in the configuration have significant effects on the aeroelastic stability of the rotors. Therefore, it is necessary to have accurate analytical methods to predict the natural vibration and aeroelastic stability characteristics of rotor blades including the multiple load paths.

The determination of natural vibration characterisitcs is basic to any dynamic design of the rotor system. Also, they form the basis for almost all practical aeroelastic stability analyses of rotor blades. Almost all the current structural dynamic programs of single load path rotor blades employ the transfer matrix method because the method is ideally suited for one-dimensional chain-like structures. In this report, this method is extended to multiple load path rotor blades without resorting to the equivalent blade modeling. Any equivalent single load path approximation may not simulate completely the dynamic behavior of multiple load path systems. With the present extension, the current rotor dynamic programs can be modified with relative ease to account for the multiple load paths without resorting to the equivalent single load path approximation. The results obtained by the transfer matrix formulation are validated by comparing with the finite-element solutions. The finite-element solutions are generated by adding a differential stiffness matrix associated with the rotation to the non-rotating stiffness matrix of the blade. To facilitate these calculations, a general purpose differential stiffness matrix due to rotation is derived analytically. This matrix will be very useful both for the generation of finite-element solutions and validation of the transfer matrix solutions.

II. BASIC EQUATIONS

The nonlinear equations of motion for the elastic bending and torsion of rotor blades are given below from Ref. 8. For algebraic conciseness, the terms e_A, B_1^*, B_2^*, C_1, C_1^* and k_A are treated as zero. However, they do not affect the general nature of the formulation presented here. Also, the aerodynamic terms L_u, L_v, and L_w are omitted from the equations of motion. To account for the differential axial displacements in the load paths the axial degree-of-freedom is included in the equations of motion.

$$
\begin{aligned}
&- \{EA(u' + \frac{v'^2}{2} + \frac{w'^2}{2})\}' - 2\Omega m\dot{v} \\
&\quad + m\ddot{u} - \Omega^2 mu = \Omega^2 mx
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&\underline{- (Tv')'} + \{[EI_{z'} \cos^2(\theta+\phi) + EI_{y'} \sin^2(\theta+\phi)]\, v'' \\
&+ (EI_{z'} - EI_{y'}) \cos(\theta+\phi) \sin(\theta+\phi) w''\}'' \\
&+ \underline{2\Omega m\dot{u}} + m\ddot{v} - me\ddot{\phi} \sin\theta - 2me\Omega(\dot{v}' \cos\theta + \dot{w}' \sin\theta) \\
&- m\Omega^2 [v + e \cos(\theta+\phi)] - 2m\Omega\beta_{pc}\dot{w} \\
&- \{me [\Omega^2 x \cos(\theta+\phi) + 2\Omega\dot{v} \cos\theta]\}' = 0
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
&\underline{- (Tw')} + \{(EI_{z'} - EI_{y'}) \cos (\theta+\phi) \sin (\theta+\phi)\, v'' \\
&+ [EI_{z'} \sin^2 (\theta+\phi) + EI_{y'} \cos^2 (\theta+\phi)]\, w''\}'' \\
&+ m\ddot{w} + me\ddot{\phi} \cos\theta + 2m\Omega\beta_{pc}\, \dot{v} \\
&- \{me [\Omega^2 x \sin (\theta+\phi) + 2\Omega\dot{v} \sin\theta]\}' = - m\Omega^2\beta_{pc}\, x
\end{aligned}
\tag{3}
$$

$$-(GJ\phi')' + (EI_{z'} - EI_{y'})[(w''^2 - v''^2)\cos\theta\sin\theta$$
$$+ v''w''\cos 2\theta] + mk_m^2\ddot{\phi} + m\Omega^2\phi\,(k_{m_2}^2 - k_{m_1}^2)\cos 2\theta$$
$$+ me\,[\Omega^2 x\,(w'\cos\theta - v'\sin\theta) - (\ddot{v} - \Omega^2 v)\sin\theta$$
$$+ \ddot{w}\cos\theta] = -\Omega^2 m\,(k_{m_2}^2 - k_{m_1}^2)\cos\theta\sin\theta - me\Omega^2\beta_{pc}\,x\cos\theta \quad (4)$$

where
$$T = EA\left(u' + \frac{v'^2}{2} + \frac{w'^2}{2}\right) \quad (5)$$

Equations (1) to (4) are coupled nonlinear partial differential equations in variables u, v, w and ϕ. Substitute Eq. (5) into Eq. (1) and integrate as shown below

$$T(x) = \int_o^x T'(x)dx = -\int_o^x \Omega^2 mxdx - \int_o^x m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)dx + k_1 + k_2 \quad (6)$$

where k_1 and k_2 are arbitrary constants.

The boundary condition for tension T(x) is given by

$$T(R) = 0 \quad (7)$$

Substitute Eq. (6) into Eq. (7)

$$-\int_o^R \Omega^2 mxdx - \int_o^R m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)dx + k_1 + k_2 = 0 \quad (8)$$

To satisfy Eq. (8), take k_1 and k_2 as

$$k_1 = \int_o^R \Omega^2 mxdx \quad (9)$$

$$k_2 = \int_o^R m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)dx \quad (10)$$

Substitute Eqs. (9) and (10) into Eq. (6)

$$T(x) = \int_x^R \Omega^2 mx\,dx + \int_x^R m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)\,dx \tag{11}$$

From Eq. (5), u' is given by

$$u' = \frac{T}{EA} - \frac{v'^2}{2} - \frac{w'^2}{2} \tag{12}$$

Substitute Eq. (11) into Eq. (12)

$$u'(x) = \frac{\Omega^2}{EA} \int_x^R mx\,dx + \frac{1}{EA} \int_x^R m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)\,dx - \frac{v'^2}{2} - \frac{w'^2}{2} \tag{13}$$

Integrate Eq. (13) with respect to x

$$u(x) = \frac{\Omega^2}{EA} \int_o^x \int_x^R mx\,dx + \frac{1}{EA} \int_o^x \int_x^R m(2\Omega\dot{v} - \ddot{u} + \Omega^2 u)\,dx$$
$$- \frac{1}{2} \int_o^x (v'^2 + w'^2)\,dx \tag{14}$$

The constant integration is zero because u(0) = 0. Differentiate Eq. (14) with respect to time

$$\dot{u} = \frac{1}{EA} \int_o^x \int_x^R m(2\Omega\ddot{v} - \dddot{u} + \Omega^2 \dot{u})\,dx - \int_o^x (v'\dot{v}' + w'\dot{w}')\,dx \tag{15}$$

Now, by substituting Eqs. (11) and (15) into Eqs. (2) and (3) for the underlined terms, two equations can be obtained in terms of v, w and ϕ. These two equations plus Eq. (4) are the necessary equations to solve for the unknowns v, w and ϕ. Once v and w are known, u can be obtained from Eq. (15) and T(x) can be obtained from Eq. (11). In essence, the bending and torsional equations are decoupled from the axial equation by the above formulation.

For determination of natural vibration characteristics, one is interested in the linear, homogeneous, undamped equations of motion. The process of obtaining these equations involves the following steps:

1. Substitution of Eqs. (11) and (15) into Eqs. (2) and (3).
2. Substitution of the following relations for small values of ϕ.

$$\cos(\theta + \phi) = \cos\theta - \phi \sin\theta$$

$$\cos^2(\theta + \phi) = \cos^2\theta - \phi \sin 2\theta$$

$$\sin(\theta + \phi) = \sin\theta + \phi \cos\theta$$

$$\sin^2(\theta + \phi) = \sin 2\theta + \phi \cos 2\theta$$

3. Dropping of nonlinear terms, that is, terms containing the products of u, v, w, ϕ and/or their derivatives.
4. Dropping of damping type terms, that is, terms containing $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{\phi}$ and their derivatives.
5. Dropping of nonhomogeneous terms, that is, terms not containing u, v, w, ϕ and/or their derivatives.

The above steps reduces Eqs. (1) to (4) to the following equations for simple harmonic time dependency of u, v, w and ϕ with frequency ω.

$$(EAu')' + (\omega^2 + \Omega^2)mu = 0 \tag{16}$$

$$\begin{aligned} &- (Tv')' + \{(EI_z\cos^2\theta + EI_y\sin^2\theta)v'' \\ &+ (EI_z - EI_y) \cos\theta \sin\theta\, w''\}'' - \omega^2\, m\, v \\ &+ \omega^2\, me\phi \sin\theta - m\Omega^2\, (v - e \sin\theta\phi) \\ &+ (me\Omega^2 x \sin\theta\phi)' = 0 \end{aligned} \tag{17}$$

$$\begin{aligned} &- (Tw')' + \{(EI_z - EI_y) \cos\theta \sin\theta\, v'' \\ &+ (EI_z \sin^2\theta + EI_y \cos^2\theta)w''\}'' - \omega^2\, mw \\ &- \omega^2\, me \cos\theta\phi - (me\Omega^2 x \cos\theta\phi)' = 0 \end{aligned} \tag{18}$$

$$- (GJ\phi')' - \omega^2 mk_m^2\phi + \Omega^2 m\,(k_{m_2}^2 - k_{m_1}^2)\cos 2\theta\phi$$

$$+ me\Omega^2 x\,(\cos\theta w' - \sin\theta v') + me\,(\omega^2 + \Omega^2)\sin\theta v$$

$$- \omega^2 me\cos\theta w = 0 \tag{19}$$

where

$$T = \Omega^2 \int_x^R mxdx \tag{20}$$

The above equations can be reduced to the following twelve non-dimensional first-order differential equations:

$$d\bar{u}/d\bar{x} = (EI_{y_o}/EAR^2)\bar{N} \tag{21}$$

$$d\bar{w}/d\bar{x} = \bar{\psi} \tag{22}$$

$$d\bar{v}/d\bar{x} = \bar{\nu} \tag{23}$$

$$d\bar{\psi}/d\bar{x} = \bar{C}_{12}\,\bar{M}_z + \bar{C}_{11}\,\bar{M}_y \tag{24}$$

$$d\bar{\nu}/d\bar{x} = \bar{C}_{22}\,\bar{M}_z + \bar{C}_{21}\,\bar{M}_y \tag{25}$$

$$d\bar{\phi}/d\bar{x} = (EI_{y_o}/GJ)\,\bar{M}_x \tag{26}$$

$$\begin{aligned} d\bar{M}_x/d\bar{x} = &- \bar{\omega}^2\bar{m}\bar{e}\cos\theta\,\bar{w} + (\bar{\omega}^2 + \bar{\Omega}^2)\,\bar{m}\bar{e}\sin\theta\,\bar{v} \\ &+ \bar{\Omega}^2\,\bar{m}\bar{x}\bar{e}\cos\theta\bar{\psi} - \bar{\Omega}^2\,\bar{m}\bar{x}\bar{e}\sin\theta\,\bar{\nu} \\ &+ \{\bar{\Omega}^2\,\bar{m}\,(\bar{k}_{m_2}^2 - \bar{k}_{m_1}^2)\cos 2\theta - \bar{\omega}^2\bar{m}\bar{k}_{m_2}^2\}\bar{\phi} \end{aligned} \tag{27}$$

$$d\bar{M}_z/d\bar{x} = \bar{T}\bar{\nu} - \bar{\Omega}^2\,\bar{m}\bar{e}\bar{x}\sin\theta\bar{\phi} - \bar{V}_y \tag{28}$$

$$d\bar{M}_y/d\bar{x} = - \bar{T}\bar{\psi} - \bar{\Omega}^2\,\bar{m}\bar{e}\bar{x}\cos\theta\bar{\phi} + \bar{V}_z \tag{29}$$

$$d\bar{V}_y/d\bar{x} = - (\bar{\omega}^2 + \bar{\Omega}^2)\,\bar{m}\bar{v} + (\bar{\omega}^2 + \bar{\Omega}^2)\,\bar{m}\bar{e}\sin\theta\bar{\phi} \tag{30}$$

$$d\bar{V}_z/d\bar{x} = - \bar{\omega}^2\,\bar{m}\bar{w} - \bar{\omega}^2\,\bar{m}\bar{e}\cos\theta\bar{\phi} \tag{31}$$

$$d\bar{N}/d\bar{x} = - (\bar{\omega}^2 + \bar{\Omega}^2)\,\bar{m}\bar{u} \tag{32}$$

where

$$\bar{x} = x/R \ ; \quad \bar{u} = u/R \ ; \quad \bar{v} = v/R \ ; \quad \bar{w} = w/R \ ; \quad \bar{\psi} = \psi \ ; \quad \bar{\nu} = \nu \ ;$$

$$\bar{\phi} = \phi \ ; \quad \bar{M}_x = M_x R/EI_{y_o} \ ; \quad \bar{M}_y R/EI_{y_o} \ ; \quad \bar{M}_z = M_z R/EI_{y_o} \ ;$$

$$\bar{N} = NR^2/EI_{y_o} \ ; \quad \bar{V}_y = V_y R^2/EI_{y_o} \ ; \quad \bar{V}_z = V_z R/EI_{y_o} \ ; \quad \bar{m} = m/m_o \ ;$$

$$\bar{e} = e/R \ ; \quad \bar{k}^2_{m_1} = k^2_{m_1}/R^2 \ ; \quad \bar{k}^2_{m_2} = k^2_{m_2}/R^2 \ ; \quad \bar{k}^2{}_m = k^2_m/R^2 \ ;$$

$$\bar{\omega}^2 = \omega^2 m_o R^4/EI_{y_o} \ ; \quad \bar{\Omega}^2 = \Omega^2 m_o R^4/EI_{y_o} \ ; \quad \bar{C}_{11} = C_{11}\ EI_{y_o} \ ;$$

$$\bar{C}_{12} = C_{12}\ EI_{y_o} \ ; \quad \bar{C}_{22} = C_{22}\ EI_{y_o} \ ; \quad \bar{C}_{21} = C_{21}\ EI_{y_o}$$

EI_{y_o} = Reference Stiffness ; m_o = Reference Mass

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1}$$

$$a_{11} = -\ (EI_y \cos^2\theta + EI_z \sin^2\theta)$$

$$a_{12} = -\ a_{21} = (EI_y - EI_z) \cos\theta \sin\theta$$

$$a_{22} = (EI_y \sin^2\theta + EI_z \cos^2\theta)$$

$$\bar{T} = \int_{\bar{x}}^{1} \bar{\Omega}^2 \bar{m}\bar{x} d\bar{x}$$

Equations (21) to (32) can be arranged into a matrix differential equation of the following form:

$$\{\bar{z}'(\bar{x})\} = [A(\bar{x})]\ \{\bar{z}(\bar{x})\} \tag{33}$$

The transfer matrix relating the state vector at any location to the initial state vector is defined by

$$\{\bar{z}(\bar{x})\} = [\bar{T}(\bar{x})]\ \{\bar{z}(0)\} \tag{34}$$

It can be shown that transfer matrix defined in the the above equation is governed by the following equation (Ref. 9):

$$[\bar{T}'(\bar{x})] = [A(\bar{x})][\bar{T}(\bar{x})] \tag{35}$$

$$[\bar{T}(0)] = [\ulcorner 1 \lrcorner], \text{ identity matrix} \tag{36}$$

where $[A(\bar{x})]$ is defined in Eq. (33).

III. NATURAL VIBRATION CHARACTERISTICS

Almost all the practical bearingless rotor designs include multiple load paths. For instance, the bearingless main rotor that was flight tested by the Boeing Vertol has three load paths, viz., two fiberglass flexbeams and a filament-wound graphite torque tube (Ref. 7). The flexbeams and torque tube are connected to the blade through a rigid clevis. The idealization of a three load path rotor blade is shown in Fig. 1.

3.1. Equilibrium Across the Clevis:

Consider the free-body diagram for the clevis as shown in Fig. 2. Let (h_{y_i}, h_{z_i}) be the location of the ith load path with reference to a coordinate system located at the blade (point 'O').

Force equilibrium requires the following relations to be satisfied:

$$N_2 = \sum_{i=1}^{n} N_{i2} \tag{37}$$

$$V_{y_2} = \sum_{i=1}^{n} V_{y_{i2}} \tag{38}$$

$$V_{z_2} = \sum_{i=1}^{n} V_{z_{i2}} \tag{39}$$

Moment equilibrium requires the following equation to be satisfied.

$$(\underset{\sim}{i}\, M_{x_2} + \underset{\sim}{j}\, M_{y_2} + \underset{\sim}{k}\, M_{z_2}) - \sum_{i=1}^{n} (\underset{\sim}{i}\, M_{x_{i2}} + \underset{\sim}{j}\, M_{y_{i2}} + \underset{\sim}{k}\, M_{z_{i2}})$$

$$- \sum_{i=1}^{n} (\underset{\sim}{j}\, h_{y_i} + \underset{\sim}{k}\, h_{z_i}) \times (\underset{\sim}{i}\, N_{i2} + \underset{\sim}{j}\, V_{y_{i2}} + \underset{\sim}{k}\, V_{z_{i2}})$$

The above equation can be written in the component form as shown below:

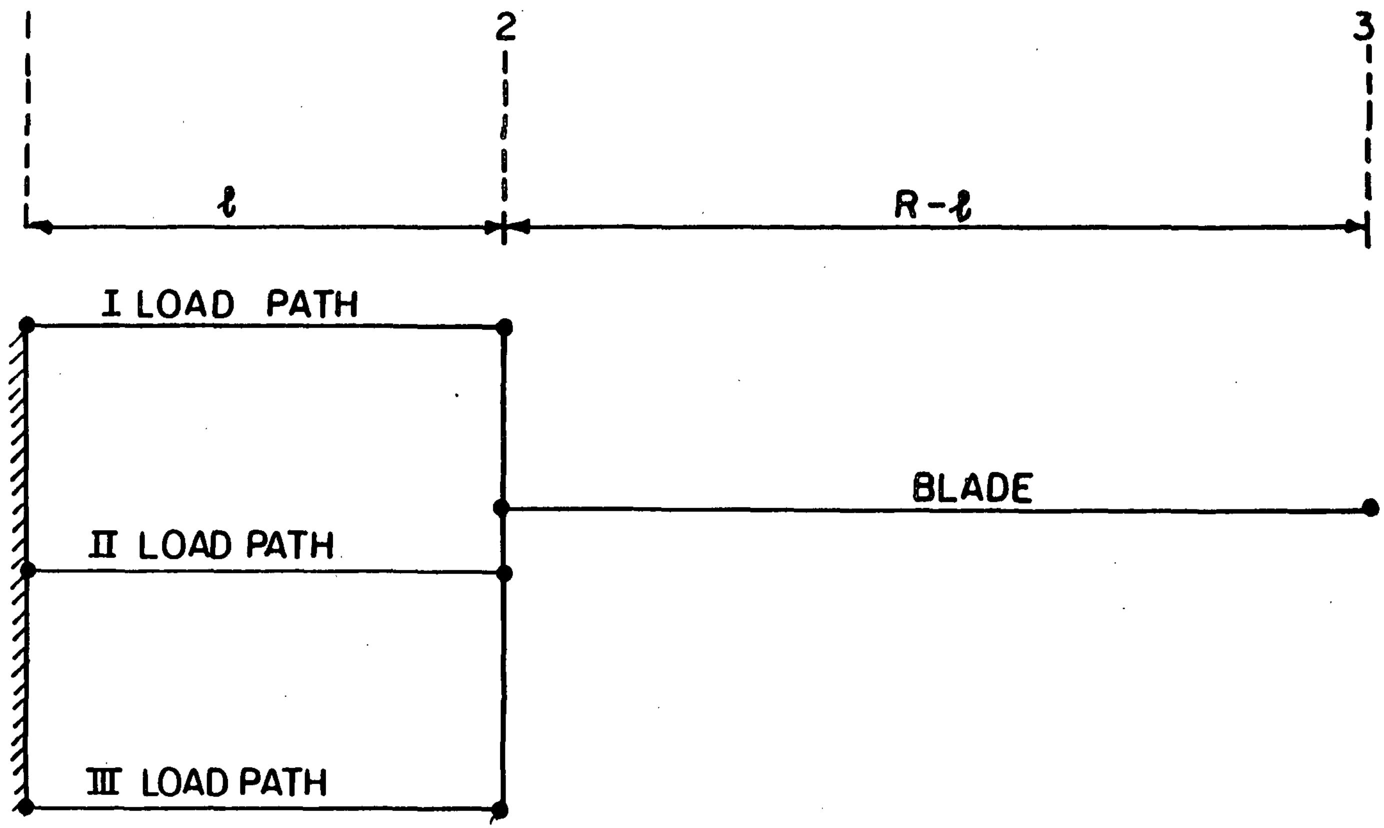
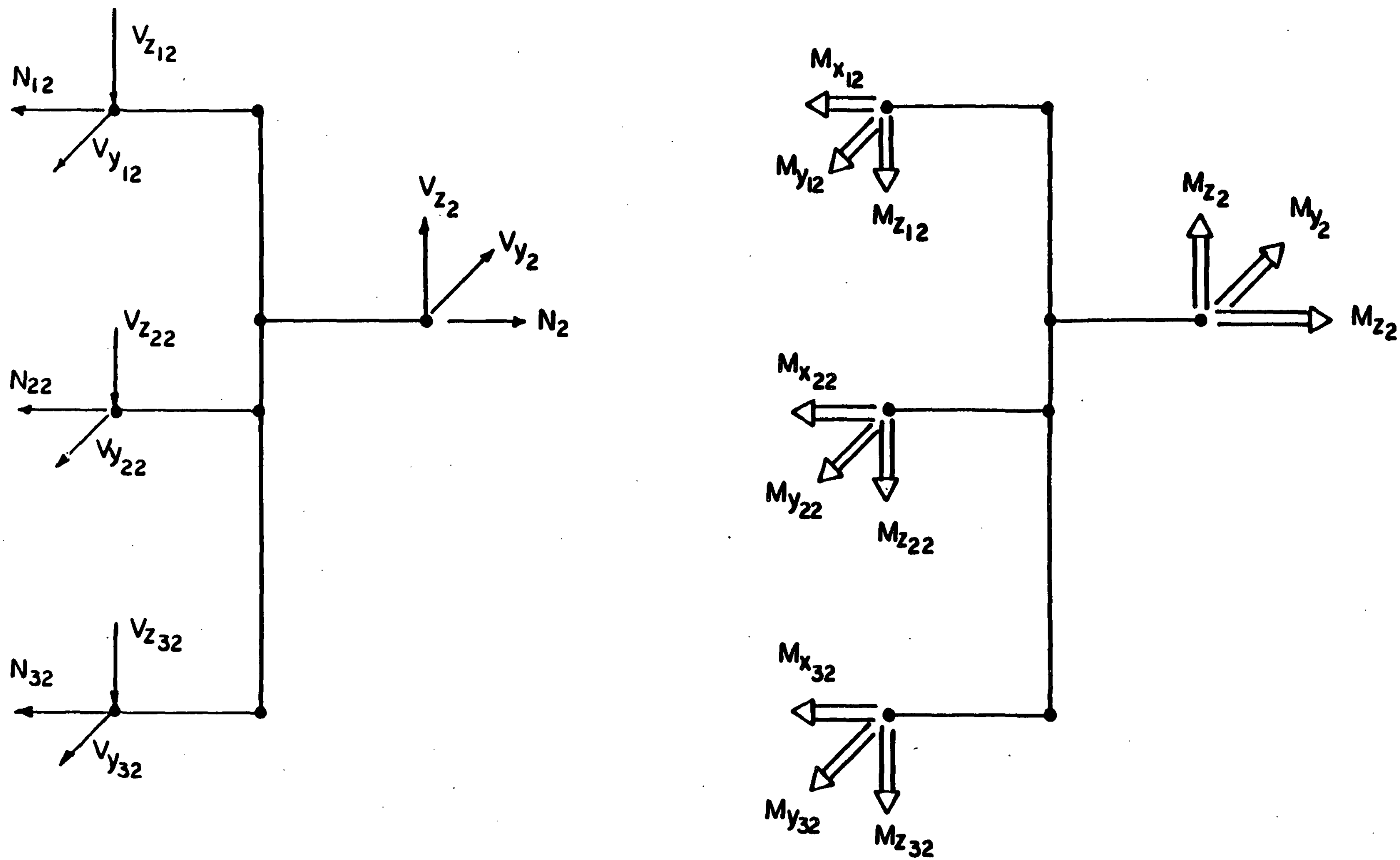


FIG. 1 MODEL FOR A TRIPPLE LOAD PATH BLADE



A. Forces

B. Moments

FIG. 2 FREE-BODY DIAGRAM OF THE CLEVIS

$$M_{x_2} = \sum_{i=1}^{n} (M_{x_{i2}} - h_{z_i} V_{y_{i2}} + h_{y_i} V_{z_{i2}}) \tag{40}$$

$$M_{y_2} = \sum_{i=1}^{n} M_{y_{i2}} + h_{z_i} N_{i2} \tag{41}$$

$$M_{z_2} = \sum_{i=1}^{n} M_{z_{i2}} - h_{y_i} N_{i2} \tag{42}$$

Equations (37) to (42) can be arranged into a matrix equation as shown below.

$$\{f_2\} = \sum_{i=1}^{n} [A_i] \{f_{i2}\} \tag{43}$$

where

$$\{f_2\}^T = \lfloor M_{x_2} \; M_{z_2} \; M_{y_2} \; V_{y_2} \; V_{z_2} \; N_2 \rfloor$$

$$\{f_{i2}\}^T = \lfloor M_{x_{i2}} \; M_{z_{i2}} \; M_{y_{i2}} \; V_{y_{i2}} \; V_{z_{i2}} \; N_{i2} \rfloor$$

and

$$[A_i] = \begin{bmatrix} 1 & 0 & 0 & -h_{z_i} & h_{y_i} & 0 \\ 0 & 1 & 0 & 0 & 0 & -h_{y_i} \\ 0 & 0 & 1 & 0 & 0 & h_{z_i} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2. Compatibility Across the Clevis

Consider the plane of the clevis as shown in Fig. 3. Let (x,0,0) be the location of the blade and (x, h_{y_i}, h_{z_i}) be the location of the ith load path before deformation. Assume that the plane of the clevis is rigid,

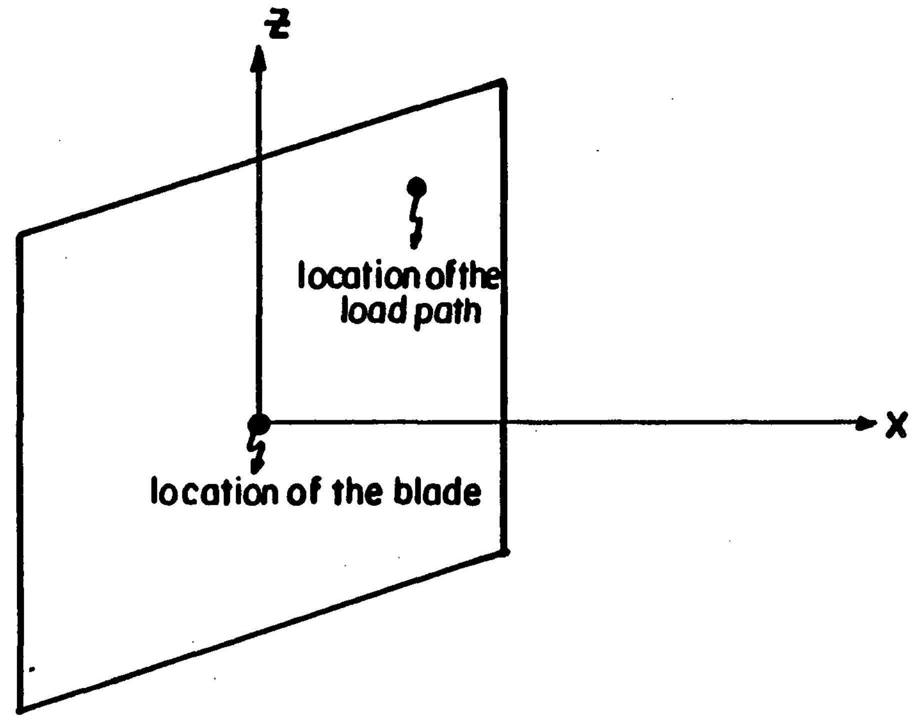


FIG. 3 PLANE OF THE CLEVIS

that is, rotates about x, y and z axes without any elastic deformation. After deformation, the x-coordinates of the blade and the ith load path are given by the second line of the following table.

State	Axial Coordinate	
	Blade	ith Load Path
Before Deformation	x	x
After Deformation	$x + u_2$	$x + u_2 - h_{z_i} \psi_2 - h_{y_i} \nu_2$

The axial displacement of ith load path is given by subtracting the x-coordinate before deformation from the x-coordinate after deformation as shown below.

$$u_{i2} = (x + u_2 - h_{z_i} \psi_2 - h_{y_i} \nu_2) - x$$

Rearrangement of the above equation yields

$$u_2 = u_{i2} + h_{z_i} \psi_{i2} + h_{y_i} \nu_{i2} \tag{44}$$

The other compatibility conditions consistent with the rigid clevis are

$$w_2 = w_{i2} \ ; \quad v_2 = v_{i2} \ ; \quad \psi_2 = \psi_{i2} \ ; \quad \nu_2 = \nu_{i2}$$

The above equations together with Eq. (44) can be arranged into a matrix equation of the following form

$$\begin{Bmatrix} u_2 \\ w_2 \\ v_2 \\ \psi_2 \\ \nu_2 \\ \phi_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & h_{z_i} & h_{y_i} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{i2} \\ w_{i2} \\ v_{i2} \\ \psi_{i2} \\ \nu_{i2} \\ \phi_{i2} \end{Bmatrix}$$

or

$$\{d_2\} = [B_i]\,\{d_{i2}\} \tag{45}$$

3.3 Frequency Determinant

Define the following relations between the state vectors from Fig. 1 and the definition of transfer matrix

$$\{z_3\} = [T]\,\{z_2\} \tag{46}$$

Equation (46) relates the state vectors of the blade at Stations 2 (clevis) and 3 (tip). Rewrite this equation into the following partitioned form.

$$\begin{Bmatrix} d_3 \\ \hline f_3 \end{Bmatrix} = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline T_{21} & T_{22} \end{array}\right] \begin{Bmatrix} d_2 \\ \hline f_2 \end{Bmatrix} \tag{47}$$

where d stands for deflections and f stands for forces.

Extract the following equation for forces from Eq. (47)

$$\{f_3\} = [T_{21}]\,\{d_2\} + [T_{22}]\,\{f_2\} \tag{48}$$

Similarly, the transfer matrix relation for the ith load path can be written as (see Fig. 1)

$$\{z_{i2}\} = [T^i]\ \{z_{i1}\} \tag{49}$$

Rewrite the above equation into partitioned form as

$$\begin{Bmatrix} d_{i2} \\ \hline f_{i2} \end{Bmatrix} = \left[\begin{array}{c|c} T^i_{11} & T^i_{12} \\ \hline T^i_{21} & T^i_{22} \end{array}\right] \begin{Bmatrix} d_{i1} \\ \hline f_{i1} \end{Bmatrix} \tag{50}$$

The displacement and force vectors can be written in terms of $\{f_{i1}\}$ by virtue of boundary condition $\{d_{i1}\} = \{0\}$ as shown below.

$$\{f_{i2}\} = [T^i_{22}]\ \{f_{i1}\} \tag{51}$$

$$\{d_{i2}\} = [T^i_{12}]\ \{f_{i1}\} \tag{52}$$

From Eq. (52)

$$\{f_{i1}\} = [T^i_{12}]^{-1}\ \{d_{i2}\} \tag{53}$$

Substitute Eq. (45) into Eq. (53)

$$\{f_{i1}\} = [T^i_{12}]^{-1}\ [B_i]^{-1}\ \{d_2\} \tag{54}$$

Substitute Eq. (54) into Eq. (51)

$$\{f_{i2}\} = [T^i_{22}]\ [T^i_{12}]^{-1}\ [B_i]^{-1}\ \{d_2\} \tag{55}$$

Substitute Eq. (55) into Eq. (43)

$$\{f_2\} = \sum_{i=1}^{n}\ [A_i]\ [T^i_{22}]\ [T^i_{12}]^{-1}\ [B_i]^{-1}\ \{d_2\} \tag{56}$$

Substitute Eq. (56) into Eq. (48)

$$\{f_3\} = ([T_{21}] + [T_{22}] \sum_{i=1}^{n} [A_i]\ [T_{22}^{i}]\ [T_{12}^{i}]^{-1}\ [B_i]^{-1})\ \{d_2\} \tag{57}$$

The vector $\{f_3\} = \{0\}$ by virtue of the boundary conditions and for nontrivial solutions of $\{d_2\}$, the determinant of the coefficient matrix should be zero and this condition yields the following frequency equation to determine the natural frequencies.

$$\det\ ([T_{21}] + [T_{22}] \sum_{i=1}^{n} [A_i]\ [T_{22}^{i}]\ [T_{12}^{i}]^{-1}\ [B_i]^{-1}) = 0 \tag{58}$$

3.4 Mode Shapes

The state vector at the clevis from Eq. (47) can be written as

$$\left\{\begin{array}{c} d_2 \\ \hline f_2 \end{array}\right\} = \left[\begin{array}{c|c} \bar{T}_{11} & \bar{T}_{12} \\ \hline \bar{T}_{21} & \bar{T}_{22} \end{array}\right] \left\{\begin{array}{c} d_3 \\ \hline f_3 \end{array}\right\} \tag{59}$$

where

$$\left[\begin{array}{c|c} \bar{T}_{12} & \bar{T}_{12} \\ \hline \bar{T}_{21} & \bar{T}_{22} \end{array}\right] = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline T_{21} & T_{22} \end{array}\right]^{-1}$$

From the above equation, the displacement vectors at the clevis and the tip of the blade can be related as shown below by virtue of the boundary condition $\{f_3\} = \{0\}$.

$$\{d_2\} = [\bar{T}_{11}]\ \{d_3\} \tag{60}$$

Substitute Eq. (60) into Eq. (57)

$$\{f_3\} = [C]\ \{d_3\} \tag{61}$$

where

$$[C] = ([T_{21}] + [T_{22}] \sum_{i=1}^{n} [A_i]\ [T_{22}^{i}]\ [T_{12}^{i}]^{-1}\ [B_i])\ [\bar{T}_{11}]$$

Assume w_3 = 1 arbitrarily and rewrite rows 2 to 6 of Eq. (61) as

$$\begin{bmatrix} C_{21} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{Bmatrix} u_3 \\ v_3 \\ \psi_3 \\ \nu_3 \\ \phi_3 \end{Bmatrix} = \begin{Bmatrix} -C_{22} \\ -C_{32} \\ -C_{42} \\ -C_{52} \\ -C_{62} \end{Bmatrix} \tag{62}$$

By solving the above equation, u_3, v_3, ψ_3, ν_3 and ϕ_3 are known and together with w_3 = 1 the entire $\{d_3\}$ is known. Once, $\{d_3\}$ is known, the state vector at the clevis can be determined from Eq. (59) as shown below

$$\begin{Bmatrix} d_2 \\ \hline f_2 \end{Bmatrix} = \begin{bmatrix} \bar{T}_{11} \\ \hline \bar{T}_{21} \end{bmatrix} \{d_3\} \tag{63}$$

Once the state vector at the clevis is known, the deflection vectors in the blade and the load paths can be obtained as follows:

Blade: By definition of the transfer matrix the state vector at any location x, is given by

$$\begin{Bmatrix} d_x \\ \hline f_x \end{Bmatrix} = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline T_{21} & T_{22} \end{array}\right]_x \begin{Bmatrix} d_2 \\ \hline f_2 \end{Bmatrix} \tag{64}$$

From the above equation

$$\{d_x\} = [T_{11}]_x \{d_2\} + [T_{12}]_x \{f_2\} \tag{65}$$

Load path: By definition of the transfer matrix the state vector at any location, x, in the load path is given by

$$\left\{\begin{array}{c} d_{ix} \\ \hline f_{ix} \end{array}\right\} = \left[\begin{array}{c|c} T^i_{11} & T^i_{12} \\ \hline T^i_{21} & T^i_{22} \end{array}\right]_x \left\{\begin{array}{c} d_{i1} \\ \hline f_{i1} \end{array}\right\} \tag{65}$$

From the above equation

$$\{d_{ix}\} = [T^i_{12}]_x \ \{f_{i1}\} \tag{67}$$

by virtue of the boundary condition ($\{d_{i1}\} = \{0\}$).

Substitute Eq. (54) into Eq. (67)

$$\{d_{ix}\} = [T^i_{12}]_x \ [T^i_{12}]^{-1} \ [B_i]^{-1} \ \{d_2\} \tag{68}$$

The mode shapes can be computed from Eqs. (65) and (68).

3.5 Tension Coefficients

The preceding formulation assumes that the transfer matrices for the blade and the load paths are either known or can be calculated. The transfer matrices can be calculated if all the coefficients of the differential equations of motion are known and tension appears as coefficient in these equations. The calculation of tensions in the blade is straight forward and it is calculated from Eq. (20).

The following procedure is employed for calculation of the tension coefficients in the load paths. Draw a free-body diagram for the clevis as shown in Fig. 4. The tension 'T' applied by the blade to the clevis is known from Eq. (20). From the free-body diagram the following equilibrium equation can be obtained

$$\{b\} = \sum_{i=1}^{n} [A_i] \ \{f_{i2}\} \tag{69}$$

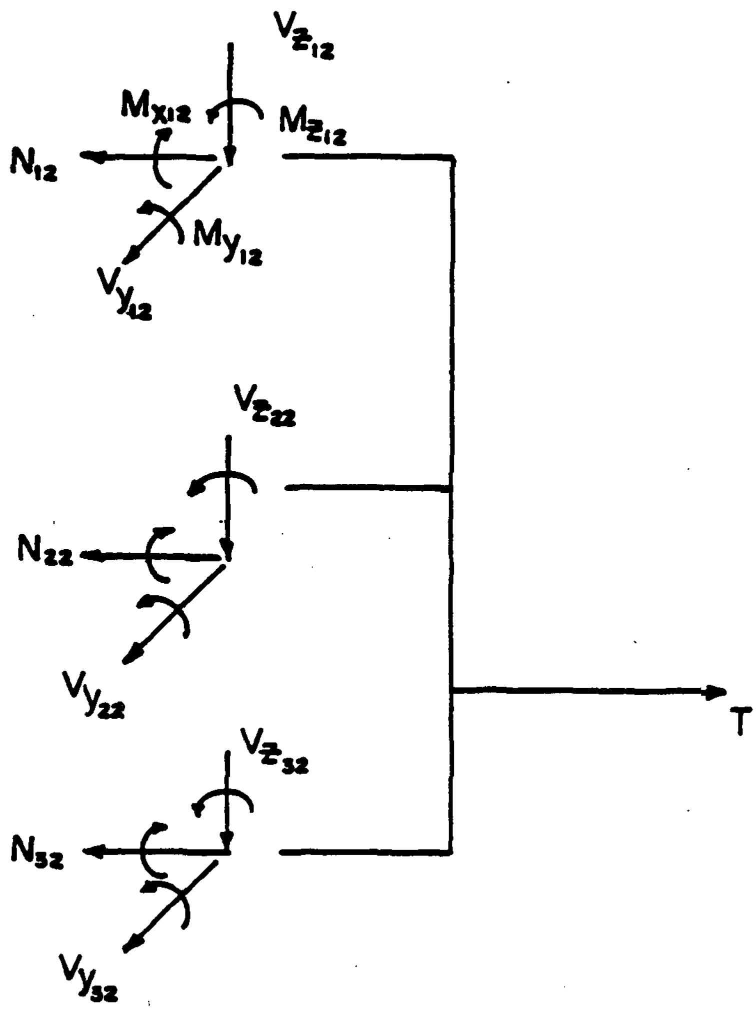


FIG. 4 TENSION CALCULATIONS IN THE LOAD PATHS

where

$$\{b\}^T = \lfloor 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad T \rfloor$$

$[A_i]$ is defined in Eq. (43)

Let $[T^i]$ be the transfer matrix of the ith load path corresponding to the static case ($\omega = 0$). By definition, one can write that

$$\begin{Bmatrix} d_{i2} \\ \hline f_{i2} \end{Bmatrix} = \left[\begin{array}{c|c} T^i_{11} & T^i_{12} \\ \hline T^i_{21} & T^i_{22} \end{array}\right] \begin{Bmatrix} d_{i1} \\ \hline f_{i1} \end{Bmatrix} \tag{70}$$

From the above equation

$$\{f_{i2}\} = [T^i_{22}] \{f_{i1}\} \tag{71}$$

Substitute Eq. (71) into Eq. (69)

$$\{b\} = \sum_{i=1}^{n} [A_i] [T^i_{22}] \{f_{i1}\} \tag{72}$$

From Eq. (70)

$$\{d_{i2}\} = [T^i_{12}] \{f_{i1}\} \tag{73}$$

by virtue of the boundary condition ($\{d_{i1}\} = \{0\}$).

Substitute Eq. (73) into Eq. (72)

$$\{b\} = \sum_{i=1}^{n} [A_i] [T^i_{22}] [T^i_{12}]^{-1} \{d_{i2}\} \tag{74}$$

Substitute Eq. (45) into Eq. (74)

$$\{b\} = \sum_{i=1}^{n} [A_i] [T^i_{22}] [T^i_{12}]^{-1} [B_i]^{-1} \{d_2\} \tag{75}$$

From the above equation

$$\{d_2\} = [D]^{-1} \{b\} \tag{76}$$

where

$$[D] = \sum_{i=1}^{n} [A_i] [T_{22}^{i}] [T_{12}^{i}]^{-1} [B_i]^{-1}$$

From Eq. (45)

$$\{d_{i2}\} = [B_i]^{-1} \{d_2\} \tag{77}$$

From Eq. (73)

$$\{f_{i1}\} = [T_{12}^{i}]^{-1} \{d_{i2}\} \tag{78}$$

Substituting Eqs. (76), (77) and (78) into Eq. (71)

$$\{f_{i2}\} = [T_{22}^{i}] [T_{12}^{i}]^{-1} [B_i]^{-1} [D]^{-1} \{b\} \tag{79}$$

The last or sixth row in the above equation is N_{i2}. Once N_{i2} is known, the tension in the ith load path can be obtained from the following equation

$$T_i(x) = \Omega^2 \int_x^{\ell} mx \, dx + N_{i2} \tag{80}$$

where

ℓ = span of the load path.

The above formulation assumes that the transfer matrices (static) of the load paths are known which depend on the tensions to be calculated. The following iterative scheme is used to solve the problem. The initial tensions in the load paths are calculated by assuming that the load paths

are coincident with the blade, i.e., $h_{y_i} = h_{z_i} = 0$. The tensions corresponding to this case are calculated as follows. The transfer matrix for axial motion of a beam for the static case is given by

$$[T(x)] = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \tag{81}$$

where

$$a = \int_0^x \frac{dx}{EA}$$

By definition of the transfer matrix

$$\begin{Bmatrix} u_{i2} \\ N_{i2} \end{Bmatrix} = \begin{bmatrix} 1 & a_i \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{i1} = 0 \\ N_{i1} \end{Bmatrix} \tag{82}$$

From the above equation

$$\left.\begin{aligned} u_{i2} &= a_i \, N_{i1} \\ N_{i2} &= N_{i1} \end{aligned}\right\} \tag{83}$$

Consider the following two cases:

Case 1: 2 Load Paths

Expanding Eq. (83)

$$\left.\begin{aligned} u_{12} &= a_1 \, N_{12} \\ u_{22} &= a_2 \, N_{22} \end{aligned}\right\} \tag{84}$$

For coincident nodes ($u_{12} = u_{22}$), Eq. (84) becomes

$$a_1 N_{12} - a_2 N_{22} = 0 \tag{85}$$

For equilibrium

$$N_{12} + N_{22} = T \tag{86}$$

By solving Eqs. (85) and (86)

$$\left.\begin{aligned} N_{12} &= \frac{a_2}{a_1 + a_2} T \\ N_{22} &= \frac{a_1}{a_1 + a_2} T \end{aligned}\right\} \tag{87}$$

Case 2: 3 Load Paths

Expanding Eq. (83) for coincident nodes

$$a_1 N_{12} = a_2 N_{22} = a_3 N_{32} \tag{88}$$

For equilibrium

$$N_{12} + N_{22} + N_{32} = T \tag{89}$$

By solving Eqs. (88) and (89)

$$\left.\begin{aligned} N_{12} &= \frac{a_2 a_3 T}{D} \\ N_{22} &= \frac{a_3 a_1 T}{D} \\ N_{32} &= \frac{a_1 a_2 T}{D} \end{aligned}\right\} \tag{90}$$

where

$$D = (a_1 a_2 + a_2 a_3 + a_3 a_1).$$

Now the above result can be generalized to the n load path case as shown below.

$$N_{i2} = \frac{\prod_{j=1}^{n,i} a_j}{\sum_{k=1}^{n} \left(\prod_{j=1}^{n,k} a_j \right)} T \tag{91}$$

where

$$\prod_{j=1}^{n,i} a_j = a_1 \, a_2 \, \ldots \, a_n / a_i$$

Now, the tension in the ith load path corresponding to the coincident load path case is given by

$$T_i^c(x) = \Omega^2 \int_o^{\ell} mx \, dx + N_{i2} \tag{92}$$

The tensions obtained from coincident nodes are used to obtain the transfer matrices of the noncoincident problem. At the most, one more iteration may be required to obtain the convergence for practical bearingless rotor designs.

IV. RESULTS

4.1 Computer Program

A general purpose computer program has been developed to determine the natural vibration characteristics of rotating multiple load path rotor blades based on the transfer matrix formulation. The listing of the computer program and the user's instructions are given in appendices A and B respectively. The significant features of the computer program are as follows.

1. The non-uniform properties including the pretwist for the blades as well as for the load paths can be handled by the program.
2. The coupled flapwise bending, chordwise bending, torsion and axial stretching degrees-of-freedom are included in the program. In the case of single load path blade the axial stretching is decoupled similar to the conventional analysis.
3. No equivalent single load path approximation is made in the program.
4. Maximum number of load paths allowed in the program is three but can be extended very easily to "n" number of load paths.
5. Continuous system model is used in the program and a fourth-order Runge-Kutta integration scheme is used to determine the transfer matrices. If a discrete model is used just like in the case of Bell Helicopter's C-81 program, the corresponding transfer matrices can be used in place of the continuous system transfer matrices and the formulation is independent of the model.

6. The natural frequencies are computed by a frequency scanning technique. Sign changes in the values of frequency determinant are detected by starting from an initial value and incrementing at steps of specified "h" until the required sign changes are detected or a final prescribed value is reached. If two frequencies are closer than the increment "h", then there is a chance of missing those frequencies. In such cases, the frequencies are detected by the fact that if any three consecutive frequency determinants have the same sign and the absolute value of the middle determinant is the smallest of the three, then there are two frequencies in that range. In this case smaller increments are taken to bracket the roots.

4.2 Numerical Results

The following data is used for numerical calculations to validate the formulation and the computer program.

1. Radius of the rotor = 260 in.
2. Distance of the clevis from the root = 52 in.
3. Length of the blade = 208 in.
4. Rotational speed (Ω) = 360 RPM
5. Flapwise bending stiffness (EI_y) = 0.2977×10^8 lb/in^2
6. Chordwise bending stiffness (EI_z) = 10×10^8 lb/in^2
7. Torsional stiffness (GJ) = 0.2×10^8 lb/in^2
8. Axial stiffness (EA) = 10^{11} lb/in^2
9. Mass per unit length = 0.0015 lb-sec^2/in^2
10. Built-in-twist = 0 deg.
11. Collective pitch = 15.026 deg.

12. e $= -0.6$ in.

13. $mk^2_{m_1}$ $= 0.89545 \times 10^{-3}$ lb-sec^2

14. $mk^2_{m_2}$ $=0.04$ lb-sec^2

For two load path blade

$h_{y_1} = 0.0$; $h_{z_1} = -1.0$ in; $h_{y_2} = 0.0$; $h_{z_2} = 3.0$ in.

For three load path blade

$h_{y_1} = 1.0$ in; $h_{z_1} = 3.0$ in; $h_{y_2} = -1.0$ in; $h_{z_2} = -1.0$ in; $h_{y_3} = 2.0$ in; $h_{z_3} = - 2.0$ in.

The natural frequencies obtained from the computer program are presented in Table 1 for load paths 1 and 2 respectively corresponding to the rotational speed $\Omega = 360$ RPM. The non-rotating and rotating ($\Omega = 360$ RPM) natural frequencies for a three load path case are presented in Table 2 and in this case out-of-plane locations are assumed for the load paths for validation of a general case. A mode shape corresponding to the two load path case is presented in Table 3. All the results obtained from the transfer matrix formulation are compared with the finite-element solutions for validation of the present formulation. The finite-element solutions are obtained from the NASTRAN program using the 80 beam elements. An external stiffness matrix associated with the centrifugal forces was computed separately and added to the non-rotating stiffness matrix calculated by the NASTRAN program. The derivation of the differential stiffness matrix associated with the blade rotation is given in the next section. From the results presented in Tables 1 to 3, it is clear that the trnasfer matrix formulation

Table 1. Comparison of natural frequencies, single and two load path cases, rad/sec

Mode No.	Single Load Path Case: Transfer matrix method	Single Load Path Case: Finite-element method	Two Load Path Case: Transfer matrix method	Two Load Path Case: Finite-element method
1	36.7738 F	36.8062 F	43.1129 F	43.1412 F
2	48.1092 C	48.1511 C	56.5803 C	56.6239 C
3	104.9309 F	104.9170 F	122.1807 F	122.1775 F
4	138.2931 T	138.2254 T	152.4027 T	152.3739 F,T
5	202.4001 F	202.3235 F	226.2707 F	226.2831 F
6	280.5927 C	280.4453 C	307.8609 C	307.7620 C
7	336.3352 F	336.1560 F	330.6756 F	330.7502 F
8	402.5505 T	402.5403 T	431.3825 T	431.3955 T
9	507.5868 F	507.2334 F	506.2530 F	506.3303 F
10	669.1642 T	667.9858 F,T	669.4276 T	668.4175 T

F = Predominantly flapwise bending

C = Predominantly chordwise bending

T = Predominantly torsion

Table 2. Comparison of natural frequencies, non-coplanar three load paths case

Mode No.	Ω = 0 Transfer matrix method	Ω = 0 Finite-element method	Ω = 360 RPM Transfer matrix method	Ω = 360 RPM Finite-element method
1	11.4292 F	11.4292 F	44.8480 F	44.8613 F
2	66.1632 C	66.2114 C	72.8896 C	73.0002 C
3	70.1983 F	70.2009 F	124.3887 F	124.4651 F
4	153.6332 T	153.6552 T	158.0232 T	157.9025 T
5	181.4035 F	181.3978 F	232.6334 F	232.6989 F
6	264.3820 F	264.4106 F	326.3415 F	326.3315 F
7	405.9868 C	406.1411 C	418.1878 C	418.1265 C
8	418.2305 F	418.1628 F	447.2150 T	447.6895 T
9	445.6527 T	445.9648 T	498.3820 C	498.3477 C
10	--	665.0317 F	669.7281 T	668.9731 T

F = Predominantly flapwise bending

C = Predominantly chordwise bending

T = Predominantly torsion

Table 3. a. Comparison of mode shapes, two load path case, third mode, Omega = 360 rpm.
(Flapwise Deflection)

x/R	Transfer Matrix Method		Finite Element Method	
	Load Path I	Load Path II	Load Path I	Load Path II
0.00	0.0	0.0	0.0	0.0
0.10	-0.0743	-0.0744	-0.0742	-0.0743
0.160	-0.1315	-0.1318	-0.1311	-0.1315
0.200	-0.1468		-0.1466	
0.312	-0.2839		-0.2835	
0.408	-0.4600		-0.4591	
0.504	-0.5732		-0.5720	
0.600	-0.5639		-0.5628	
0.712	-0.3533		-0.3521	
0.808	+0.0008		+0.0019	
0.904	+0.4747		+0.4754	
1.00	+1.0000		+1.0000	

Table 3. b. Comparison of mode shapes, two load path case, third mode, Omega = 360 rpm.
(Chordwise Deflection)

x/R	Transfer Matrix Method		Finite Element Method	
	Load Path I	Load Path II	Load Path I	Load Path II
0.00	0.0	0.0	0.0	0.0
0.10	0.0210	0.0209	0.0210	0.0210
0.160	0.0380	0.0379	0.0380	0.0381
0.200	0.0436		0.0437	
0.312	0.0835		0.0838	
0.408	0.1308		0.1313	
0.504	0.1590		0.1598	
0.600	0.1524		0.1538	
0.712	0.0890		0.0910	
0.808	0.0132		-0.0019	
0.904	-0.1482		-0.1446	
1.00	-0.2972		-0.2927	

Table 3. c. Comparison of mode shapes, two load path case, third mode, Omega = 360 rpm.
(Torsional Deflection)

x/R	Transfer Matrix Method		Finite Element Method	
	Load Path I	Load Path II	Load Path I	Load Path II
0.00	0.0	0.0	0.0	0.0
0.10	0.0015	0.0015	0.0016	0.0016
0.160	0.0023	0.0023	0.0026	0.0026
0.200	0.0029		0.0032	
0.312	0.0061		0.0064	
0.408	0.0086		0.0089	
0.504	0.0109		0.0122	
0.600	0.0126		0.0130	
0.712	0.0140		0.0145	
0.808	0.0147		0.0153	
0.904	0.0150		0.0156	
1.00	0.0151		0.0157	

yields very accurate results for multiple load path rotor blades. In fact, the slight discrepancies noticed in these results is due to the approximations made in the finite-element solutions, for instance, a constant tension is assumed within each finite-element and actual variation is used in the transfer matrix solution.

4.3 Differential Stiffness Matrix Due to Rotation

The work done by centrifugal forces due to combined flapwise bending, chordwise bending, torsion and axial stretching is given by (Ref. 10)

$$
\begin{aligned}
P = {} & \frac{1}{2}\int_o^\ell T(v'^2 + w'^2)dx - \frac{1}{2}\int_o^\ell \Omega^2 (u^2 + v^2)dx \\
& - \frac{1}{2}\int_o^\ell \Omega^2 me (\cos\beta - \phi\sin\beta)v\, dx \\
& + \int_o^\ell \Omega^2 m \{xe (v' \cos\beta + w' \sin\beta) \\
& + \phi xe (-v' \sin\beta + w' \cos\beta)\}dx \\
& + \frac{1}{2}\int_o^\ell \Omega^2 m (k_{m_2}^2 - k_{m_1}^2) (\sin 2\beta + \phi\cos 2\beta)\phi\, dx
\end{aligned}
\tag{93}
$$

Consider the following displacement functions for flapwise bending, chordwise bending, torsion and axial displacement in terms of the shape functions

$$
\begin{aligned}
w(x) = {} & w_1 (1 - 3x^2/\ell^2 + 2x^3/\ell^3) + \theta_{y_1} (-x + 2x^2/\ell - x^3/\ell^2) \\
& + w_2 (3x^2/\ell^2 - 2x^3/\ell^3) + \theta_{y_2} (x^2/\ell - x^3/\ell^2)
\end{aligned}
\tag{94}
$$

$$v(x) = v_1 \ (1 - 3x^2/\ell^2 + 2x^3/\ell) + \theta_{z_1} \ (x - 2x^2/\ell + x^3/\ell^2)$$
$$+ v_2 \ (3x^2/\ell^2 - 2x^3/\ell^3) + \theta_{z_2} \ (-x^2/\ell + x^3/\ell^2) \tag{95}$$

$$\phi(x) = \phi_1 \ (1 - x/\ell) + \phi_2 \ (x/\ell) \tag{96}$$

$$u(x) = u_1 \ (1 - x/\ell) + u_2 \ (x/\ell) \tag{97}$$

Substitute Eqs. (94) to (97) into Eq. (93) and arrange the resulting equation into the following form

$$P = \frac{1}{2} \{d\}^T \ [k] \ \{d\} \tag{98}$$

where

$$\{d\}^T = \lfloor u_1 \ v_1 \ w_1 \ \phi_1 \ \theta_{y_1} \ \theta_{z_1} \ u_2 \ v_2 \ w_2 \ \phi_2 \ \theta_{y_2} \ \theta_{z_2} \rfloor .$$

The matrix [k] in Eq. (98) is the differential stiffness matrix due to rotation. The final form of this matrix is given in Appendix C.

References

1. Fenaughty, R.R. and Noehren, W.L., "Composite Bearingless Tail Rotor for UTTAS", Journal of the American Helicopter Society, Vol. 22, (3), July 1977, pp. 19-26.

2. Staley, J.A., Gabel, R., and MacDonald, H.I., "Full Scale Ground and Air Resonance Testing of the Army - Boeing Vertol Bearingless Main Rotor", Paper No. 79-23, American Helicopter Society 35th Annual National Forum, Washington, D.C., May 1979.

3. Dixon, P.G.C. and Bishop, H.E., "The Bearingless Main Rotor", Journal of the American Helicopter Society, Vol. 25 (3), July 1980, pp. 15-21.

4. Bousman, W.G., Ormiston, R.A. and Mirick, P.H., "Design Considerations for Bearingless Main Rotor Hubs", American Helicopter Society 39th Annual National Forum, St. Louis, Missouri, May 1983.

5. Bielawa, R.L., Cheney, Jr., M.C. and Novak, R.C., "Investigation of a Bearingless Helicopter Rotor Concept Having a Composite Primary Structure", NASA CR-2637, Oct. 1976.

6. Bielawa, R.L., "Aeroelastic Analysis for Helicopter Rotor Blades with Time-Variable, Nonlinear Structural Redundancy-Mathematical Derivation and Program User's Manual", NASA CR-2638, Oct. 1976.

7. Dixon, P.G.C., "Design, Development, and Flight Demonstration of the Loads and Stability Characterisitcs of Bearingless Main Rotor", USA AVRADCOM-TR-80-D3, June 1980.

8. Hodges, D.H. and Dowell, E.H., "Nonlinear Equations of Motion for the Elastic Bending and Torsion of Twisted Nonuniform Rotor Blades," NASA TND-7818, 1974.

9. Murthy, V.R., "Dynamic Characterisitcs of Rotor Blades", Journal of Sound and Vibration, Vol. 49(4), 1976, pp. 483-500.

10. Houbolt, J.C. and Brooks, G.W., "Differential Equations of Motion for Combined Flapwise Bending, Chordwise Bending, and Torsion of Twisted Non-uniform Rotor Blades", NASA Report 1346, 1958.

APPENDIX A

```
C     *****************************************************************
C      THIS PROGRAM COMPUTES THE NATURAL VIBRATION CHARACTERISTICS OF
C      MULTIPLE LOAD PATH ROTOR BLADES (UPTO THREE LOAD PATHS)
C      UNDERGOING BENDING - BENDING - TORSION - AXIAL VIBRATIONS.
C     *****************************************************************
C
C      ---------------------------------------------------------------
C                              DECLARATION STATEMENTS
C      ---------------------------------------------------------------
C
       IMPLICIT REAL*8 (A - H, O - Z)
       REAL*8 MASS,MASS1(3,21),MASS2(101), KM1S1(3,21)
       REAL*8 KM2S1(3,21), KM1S2(101), KM2S2(101)
       DIMENSION   STA1(21),    STA2(101),  EIY1(3,21),
      1 EIY2(101), EA1(3,21),   EA2(101),   EIZ1(3,21),
      1 EIZ2(101), GJ1(3,21),   GJ2(101),   E1(3,21),
      1 E2(101),   BETA1(3,21), BETA2(101), W1(3,11),
      1 W2(51),   V1(3,11),   V2(51),    PHI1(3,11),  PHI2(51),
      1 SL1(11), SL2(51), STA(101), FD(3,2),
      1 U(3),  BB(12),  AT(3), TEN1(3,21),
      1 TEN2(101), D11(3,21),  D12(3,21),
      1 D13(3,21), D14(3,21),  D15(3,21),
      1 D16(3,21), D17(3,21),  D18(3,21),
      1 D19(3,21), D191(3,21), D192(3,21),
      1 D21(101),  D22(101),   D23(101),
      1 D24(101),  D25(101),   D26(101),
      1 D27(101),  D28(101),   D29(101),
      1 D291(101), D292(101),  FREQEN(10),
      1 TF1(3,12,12), TF2(12,12), XA(6,6), XB(6,6), XC(6,6),
      1 XD(6,6), XE(6,6), XR(6,6)
       COMMON/X1/FREQEN, H, H1, H2, IJK
       COMMON/X2/FACT
       COMMON/X3/STA, NS
       COMMON/X4/NPATH,ICPL
       COMMON/X5/EIY1,TEN2,EA1,EA2,D11,D21,D12,D22,D13,
      1   D23,D14,D24,D15,D25,D16,D26,D17,D27,D18,D28,D19,
      1   D29,D191,D291,D192,D292,OMEGAN
       COMMON/X6/SL1,SL2,CPM,FRE,HERTZ,BLANK,DOT,STAR,J1,IPLOT
       COMMON/X7/ HH1, HH2
       COMMON/X8/ BB, IND
       COMMON/X9/ FD
       COMMON/X10/ TF1, TF2
       COMMON/X11/ XA, XB, XC, XD, XE, XR
       COMMON/X12/CON
C
```

```
C      ----------------------------------------------------------------
C                 THIS SECTION READS THE DATA OF THE SYSTEM
C      ----------------------------------------------------------------
C
       READ(20,100) NPATH, ISTAGE, IPLOT, NS1, NS2
       READ(20,105) SPAN1, SPAN2, SCH, OMEGA
       READ(20,105) (STA1(I), I=1,NS1)
       READ(20,105) (STA2(I), I=1,NS2)
       DO 400 I=1,NPATH
          READ(20,105) (MASS1(I,J), J=1,NS1)
          READ(20,105) (EIY1(I,J),  J=1,NS1)
          READ(20,105) (EIZ1(I,J),  J=1,NS1)
          READ(20,105) (GJ1(I,J),   J=1,NS1)
          READ(20,105) (E1(I,J),    J=1,NS1)
          READ(20,105) (BETA1(I,J), J=1,NS1)
          READ(20,105) (KM1S1(I,J), J=1,NS1)
          READ(20,105) (KM2S1(I,J), J=1,NS1)
          READ(20,105) (EA1(I,J),   J=1,NS1)
 400   CONTINUE
       READ(20,105) (MASS2(I), I=1,NS2)
       READ(20,105) (EIY2(I),  I=1,NS2)
       READ(20,105) (EIZ2(I),  I=1,NS2)
       READ(20,105) (GJ2(I),   I=1,NS2)
       READ(20,105) (E2(I),    I=1,NS2)
       READ(20,105) (BETA2(I), I=1,NS2)
       READ(20,105) (KM1S2(I), I=1,NS2)
       READ(20,105) (KM2S2(I), I=1,NS2)
       READ(20,105) (EA2(I),   I=1,NS2)
       READ(20,105)    H1,       H,         H2
       IF(NPATH.GT.1)  READ(20,105) ((FD(I,J),J = 1,2),I = 1, NPATH)
       IF(NPATH.GT.1)  READ(20,100) ITR
       IF(ISTAGE.NE.1) READ(20,100) NF
       IF(ISTAGE.EQ.4) READ(20,105) (FREQEN(J),J = 1,NF)
       IF(IPLOT.EQ.1)  READ(20,110) BLANK, DOT, STAR
C
```

```
C     ------------------------------------------------------------
C                THIS SECTION PRINTS THE DATA OF THE SYSTEM
C     ------------------------------------------------------------
C
      SPAN = SPAN1 + SPAN2
      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,347) ISTAGE
      WRITE(22,348) IPLOT
      IF (ISTAGE .NE. 1) WRITE(22,210) NF
      WRITE(22,310) H1
      WRITE(22,215) H
      WRITE(22,220) H2
      WRITE(22,225) SPAN
      WRITE(22,230) SPAN1
      WRITE(22,235) SPAN2
      WRITE(22,236) SCH
      WRITE(22,240) OMEGA
      WRITE(22,245) NS1
      WRITE(22,250) NS2
      WRITE(22,315) NPATH
      DO 405 I = 1, NPATH
         IF(I .EQ. 1) WRITE(22,255)
         IF(I .EQ. 2) WRITE(22,260)
         IF(I .EQ. 3) WRITE(22,265)
         WRITE(22,270)
         WRITE(22,275) (STA1(J),    J=1,NS1)
         WRITE(22,280)
         WRITE(22,275) (MASS1(I,J), J=1,NS1)
         WRITE(22,285)
         WRITE(22,275) (EIY1(I,J),  J=1,NS1)
         WRITE(22,350)
         WRITE(22,275) (EIZ1(I,J),  J=1,NS1)
         WRITE(22,352)
         WRITE(22,275) (GJ1(I,J),   J=1,NS1)
         WRITE(22,354)
         WRITE(22,275) (E1(I,J),    J=1,NS1)
         WRITE(22,356)
         WRITE(22,275) (BETA1(I,J), J=1,NS1)
         WRITE(22,358)
         WRITE(22,275) (KM1S1(I,J), J=1,NS1)
         WRITE(22,360)
         WRITE(22,275) (KM2S1(I,J), J=1,NS1)
         WRITE(22,286)
         WRITE(22,275) (EA1(I,J),   J=1,NS1)
         IF(NPATH .GT. 1) WRITE(22,321) I, FD(I,1), FD(I,2)
         IF(NPATH .GT. 1) WRITE(22,375) ITR
 405  CONTINUE
      WRITE(22,290)
      WRITE(22,270)
      WRITE(22,275)  (STA2(I), I=1,NS2)
      WRITE(22,280)
      WRITE(22,275) (MASS2(I), I=1,NS2)
      WRITE(22,285)
      WRITE(22,275) (EIY2(I),  I=1,NS2)
      WRITE(22,350)
```

```
      WRITE(22,275) (EIZ2(I),  I=1,NS2)
      WRITE(22,352)
      WRITE(22,275) (GJ2(I),   I=1,NS2)
      WRITE(22,354)
      WRITE(22,275) (E2(I),    I=1,NS2)
      WRITE(22,356)
      WRITE(22,275) (BETA2(I), I=1,NS2)
      WRITE(22,358)
      WRITE(22,275) (KM1S2(I), I=1,NS2)
      WRITE(22,360)
      WRITE(22,275) (KM2S2(I), I=1,NS2)
      WRITE(22,343)
      WRITE(22,275) (EA2(I),   I=1,NS2)
      WRITE(22,205)
C
```

```
C       -------------------------------------------------------------
C                      THIS SECTION INTERPOLATES THE DATA
C       -------------------------------------------------------------
C
      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,295)
         NS = NS1
         DO 411 J = 1, NS1
         STA(J) = STA1(J)
 411     CONTINUE
         HH = SPAN1/20.0
      DO 420 I = 1, NPATH
         DO 410  J = 1, NS1
           D21(J)  = MASS1(I,J)
           D22(J)  = EIY1(I,J)
           D23(J)  = EIZ1(I,J)
           D24(J)  = GJ1(I,J)
           D25(J)  = E1(I,J)
           D26(J)  = BETA1(I,J)
           D27(J)  = KM1S1(I,J)
           D28(J)  = KM2S1(I,J)
           TEN2(J) = EA1(I,J)
 410     CONTINUE
         CALL INTPOL(21, D21, HH)
         CALL INTPOL(21, D22, HH)
         CALL INTPOL(21, D23, HH)
         CALL INTPOL(21, D24, HH)
         CALL INTPOL(21, D25, HH)
         CALL INTPOL(21, D26, HH)
         CALL INTPOL(21, D27, HH)
         CALL INTPOL(21, D28, HH)
         CALL INTPOL(21,TEN2, HH)
         DO 415 J = 1, 21
           MASS1(I,J) = D21(J)
           EIY1(I,J)  = D22(J)
           EIZ1(I,J)  = D23(J)
           GJ1(I,J)   = D24(J)
           E1(I,J)    = D25(J)
           BETA1(I,J) = D26(J)
           KM1S1(I,J) = D27(J)
           KM2S1(I,J) = D28(J)
           EA1(I,J)   = TEN2(J)
 415     CONTINUE
         IF(I .EQ. 1) WRITE(22,255)
         IF(I .EQ. 2) WRITE(22,260)
         IF(I .EQ. 3) WRITE(22,265)
         WRITE(22,280)
         WRITE(22,275) (D21(J),  J=1,21)
         WRITE(22,285)
         WRITE(22,275) (D22(J),  J=1,21)
         WRITE(22,350)
         WRITE(22,275) (D23(J),  J=1,21)
         WRITE(22,352)
         WRITE(22,275) (D24(J),  J=1,21)
         WRITE(22,354)
```

```
         WRITE(22,275) (D25(J),  J=1,21)
         WRITE(22,356)
         WRITE(22,275) (D26(J),  J=1,21)
         WRITE(22,358)
         WRITE(22,275) (D27(J),  J=1,21)
         WRITE(22,360)
         WRITE(22,275) (D28(J),  J=1,21)
         WRITE(22,286)
         WRITE(22,275) (TEN2(J), J=1,21)
 420  CONTINUE
      NS = NS2
      DO 421 J  = 1, NS2
         STA(J) = STA2(J)
 421  CONTINUE
      HH = SPAN2/100.0
      CALL INTPOL(101, MASS2, HH)
      CALL INTPOL(101, EIY2,  HH)
      CALL INTPOL(101, EIZ2,  HH)
      CALL INTPOL(101, GJ2,   HH)
      CALL INTPOL(101, E2,    HH)
      CALL INTPOL(101, BETA2, HH)
      CALL INTPOL(101, KM1S2, HH)
      CALL INTPOL(101, KM2S2, HH)
      CALL INTPOL(101, EA2,   HH)
      WRITE(22,300)
      WRITE(22,280)
      WRITE(22,275) (MASS2(J), J=1,101)
      WRITE(22,285)
      WRITE(22,275) (EIY2(J),  J=1,101)
      WRITE(22,350)
      WRITE(22,275) (EIZ2(J),  J=1,101)
      WRITE(22,352)
      WRITE(22,275) (GJ2(J),   J=1,101)
      WRITE(22,354)
      WRITE(22,275) (E2(J),    J=1,101)
      WRITE(22,356)
      WRITE(22,275) (BETA2(J), J=1,101)
      WRITE(22,358)
      WRITE(22,275) (KM1S2(J), J=1,101)
      WRITE(22,360)
      WRITE(22,275) (KM2S2(J), J=1,101)
      WRITE(22,343)
      WRITE(22,275) (EA2(J),   J=1,101)
C
```

```
C     ------------------------------------------------------------
C       THIS SECTION NONDIMENSIONALIZES THE DATA AND DETERMINES THE
C       COEFFICIENTS OF THE FIRST ORDER DIFFERENTIAL EQUATIONS
C     ------------------------------------------------------------
C
      PI     = 4.0 * DATAN(1.0D+00)
      OMEGA  = OMEGA * PI/30.0
      OMEGAS = OMEGA * OMEGA
      IF(NPATH .EQ. 1) FD(1,1) = 0.0
      IF(NPATH .EQ. 1) FD(1,2) = 0.0
      ICPL     = 0.0
      DO 422 I = 1, NPATH
      DO 422 J = 1,2
      IF(FD(I,J) .GE. 1.0E-10) ICPL = 1
 422  CONTINUE
      STEP = SPAN2/100.0
      X    = SPAN1
      H5   = STEP/24.0
      DO 425 I = 1, 101
         D21(I) = MASS2(I) * H5 * X
         X      = X + STEP
 425  CONTINUE
      TEN2(101) = 0.0
      TEN2(100) = ( 9.0 * D21(101) + 19.0 *
     1 D21(100) - 5.0 * D21(99) + D21(98) ) * OMEGAS
      DO 426 I  = 2,99
             J  = 101 - I
      TEN2(J)   = TEN2(J+1) + ( -D21(J+2) + 13.0 *
     1 ( D21(J+1) + D21(J) ) - D21(J-1) ) * OMEGAS
 426  CONTINUE
      TEN2(1)   = TEN2(2) + ( D21(4) - 5.0 *
     1 D21(3) + 19.0 * D21(2) + 9.0 * D21(1) ) * OMEGAS
      EIY       = EIY2(1)
      MASS      = MASS2(1)
      SPANS     = SPAN * SPAN
      FA        = SPANS / EIY
      FACT      = DSQRT(FA * SPANS * MASS)
      CON       = SPAN / SCH
      OMEGAN    = OMEGAS * FACT * FACT
      HH        = SPAN1 / (20.0 * SPAN)
      DO 4251 I = 1, NPATH
                X = 0.0
       DO 4251  J = 1,21
       KM1S1(I,J) = KM1S1(I,J) / MASS1(I,J)
       KM2S1(I,J) = KM2S1(I,J) / MASS1(I,J) + E1(I,J) * E1(I,J)
         EA1(I,J) = 1.0 / ( EA1(I,J) * FA )
         PITCH    = BETA1(I,J) * PI / 180.0
         CO       = DCOS(PITCH)
         SI       = DSIN(PITCH)
         CS       = CO * CO
         SS       = SI * SI
         A11      = -EIZ1(I,J) * SS - EIY1(I,J) * CS
         A12      = -CO * SI * (EIZ1(I,J) - EIY1(I,J))
         A22      = EIZ1(I,J) * CS + EIY1(I,J) * SS
         DE       = A11 * A22 + A12 * A12
         D19(I,J) = -A12 * EIY / DE
```

```
          D191(I,J)= A22 * EIY / DE
          D192(I,J)= A11 * EIY / DE
          D11(I,J) = EIY/GJ1(I,J)
          D12(I,J) = MASS1(I,J)/MASS
          D13(I,J) = D12(I,J) * CO * E1(I,J) / SPAN
          D14(I,J) = D12(I,J) * SI * E1(I,J) / SPAN
          D15(I,J) = D13(I,J) * X * OMEGAN
          D16(I,J) = D14(I,J) * X * OMEGAN
          D17(I,J) = OMEGAN * D12(I,J) * (KM2S1(I,J) -
     1 KM1S1(I,J)) * (CS - SS) / SPANS
          D18(I,J) = D12(I,J) * (KM1S1(I,J) + KM2S1(I,J)) / SPANS
                X  = X + HH
4251 CONTINUE
       X  = SPAN1 / SPAN
       HH = SPAN2 / (100.0* SPAN)
          DO 428 J = 1, 101
          KM1S2(J) = KM1S2(J) / MASS2(J)
          KM2S2(J) = KM2S2(J) / MASS2(J) + E2(J) * E2(J)
            EA2(J) = 1.0 / (EA2(J) * FA)
             PITCH = BETA2(J) * PI / 180.0
                CO = DCOS(PITCH)
                SI = DSIN(PITCH)
                CS = CO * CO
                SS = SI * SI
               A12 = -CO * SI * (EIZ2(J) - EIY2(J))
               A11 = -EIZ2(J) * SS - EIY2(J) * CS
               A22 = EIZ2(J) * CS + EIY2(J) * SS
               DE  = A11 * A22 + A12 * A12
               D29(J)  = -A12 * EIY / DE
               D291(J) = A22 * EIY / DE
               D292(J) = A11 * EIY / DE
               D21(J)  = EIY / GJ2(J)
               D22(J)  = MASS2(J) / MASS
               D23(J)  = D22(J) * CO * E2(J) / SPAN
               D24(J)  = D22(J) * SI * E2(J) / SPAN
               D25(J)  = D23(J) * X * OMEGAN
               D26(J)  = D24(J) * X * OMEGAN
               D27(J)  = OMEGAN * D22(J) * (KM2S2(J) -
     1                   KM1S2(J) ) * (CS - SS) / SPANS
               D28(J)  = D22(J) * (KM1S2(J) + KM2S2(J)) / SPANS
               TEN2(J) = FA * TEN2(J)
                     X = X + HH
428      CONTINUE
       HH1  = SPAN1 / (10.0 * SPAN)
       HH2  = SPAN2 / (50.0 * SPAN)
       STEP = SPAN1 / (20.0 * SPAN)
       H5   = STEP / 24.0
       GO TO (4281, 4282, 4282), NPATH
4281       BB(1) = TEN2(1)
       GO TO 4286
4282     DO 4284 J = 1,NPATH
             AT(J) = (9.0 * EA1(J,1) + 19.0 * EA1(J,2) - 5.0
     1               * EA1(J,3) + EA1(J,4) ) * H5
         DO 4283 K = 1,18
             AT(J) = AT(J) + H5 * ( -EA1(J,K) + 13.0 * EA1(J,K+1)
     1               + 13.0 * EA1(J,K+2) - EA1(J,K+4) )
```

```
4283 CONTINUE
           AT(J) = AT(J) + H5 * ( EA1(J,18) - 5.0 * EA1(J,19)
     1               + 9.0 * EA1(J,20) + 9.0 * EA1(J,21) )
4284 CONTINUE
      IF(NPATH .EQ. 3) GO TO 4285
      BB(1) = AT(1) * TEN2(1) / ( AT(1) + AT(2) )
      BB(2) = AT(2) * TEN2(1) / ( AT(1) + AT(2) )
      GO TO 4286
4285 PR = AT(1) * AT(2) + AT(2) * AT(3) + AT(3) * AT(1)
      BB(1) = TEN2(1) * AT(2) * AT(3) / PR
      BB(2) = TEN2(1) * AT(3) * AT(1) / PR
      BB(3) = TEN2(1) * AT(1) * AT(2) / PR
4286 CONTINUE
      WRITE(22,380)
      WRITE(22,275) ((BB(I) / FA),I = 1,NPATH)
      DO 4290 JJ = 1, NPATH
      X = 0.0
      DO 4287 I = 1,21
      MASS2(I) = MASS1(JJ,I) * H5 * X / MASS
      X = X + STEP
4287 CONTINUE
      TEN1(JJ,21) = 0.0
      TEN1(JJ,20) = (9.0 * MASS2(21) + 19.0 * MASS2(20)
     1               - 5.0 * MASS2(19) + MASS2(18) ) * OMEGAN
      DO 429 I = 2,19
      J = 21 - I
      TEN1(JJ,J) = TEN1(JJ,J+1) + ( -MASS2(J+2) + 13.0 *(MASS2(J+1)
     1              + MASS2(J)) - MASS2(J-1) ) * OMEGAN
429   CONTINUE
      TEN1(JJ,1) = TEN1(JJ,2) + ( MASS2(4) - 5.0 * MASS2(3)
     1              + 19.0 * MASS2(2) + 9.0 * MASS2(1) ) * OMEGAN
4290 CONTINUE
      IF(ICPL .EQ. 0) ITR = 1
      DO 4291 I = 1, NPATH
      DO 4291 J = 1,21
      EIY1(I,J) = TEN1(I,J) + BB(I)
4291 CONTINUE
      WRITE(22,376)
      DO 4292 I = 1,NPATH
      WRITE(22,377) I
      WRITE(22,275) ((EIY1(I,J)/FA),J = 1,21)
4292 CONTINUE
      DO 4318 IK = 1, ITR
      DO 431 J = 1,6
      DO 431 K = 1,6
       XR(J,K) = 0.0
431   CONTINUE
      CALL TRAMAT(0.0D+00)
      DO 4314 I = 1, NPATH
      DO 4310 J = 1,6
      DO 4310 K = 1,6
       XC(J,K) = TF1(I,J,K+6)
       XD(J,K) = TF1(I,J+6,K+6)
4310 CONTINUE
      CALL SOLUTN(XC,6,-1,6)
      DO 4311 J = 1,6
```

```
        XC(J,4) = XC(J,4) - FD(I,2) * XC(J,1)
        XC(J,5) = XC(J,5) - FD(I,1) * XC(J,1)
 4311 CONTINUE
      CALL MATMUL(6,6,6,XD,XC,XE)
      DO 4312 J = 1,6
      DO 4312 K = 1,6
        XC(J,K) = XE(J,K)
      IF(I .EQ. 1) XA(J,K) = XE(J,K)
      IF(I .EQ. 2) XB(J,K) = XE(J,K)
 4312 CONTINUE
      DO 4313 J = 1,6
        XE(1,J) = XE(1,J) - FD(I,2) * XE(4,J) + FD(I,1) * XE(5,J)
        XE(2,J) = XE(2,J) - FD(I,1) * XE(6,J)
        XE(3,J) = XE(3,J) + FD(I,2) * XE(6,J)
 4313 CONTINUE
      DO 4314 J = 1,6
      DO 4314 K = 1,6
        XR(J,K) = XR(J,K) + XE(J,K)
 4314 CONTINUE
      CALL SOLUTN(XR,6,-1,6)
      DO 4315 J = 1,6
        XD(J,1) = XR(J,6) * TEN2(1)
 4315 CONTINUE
      DO    4316 I = 1, NPATH
      IF(I .EQ. 1) CALL MATMUL(6,6,1,XA,XD,XE)
      IF(I .EQ. 2) CALL MATMUL(6,6,1,XB,XD,XE)
      IF(I .EQ. 3) CALL MATMUL(6,6,1,XC,XD,XE)
      BB(I) = XE(6,1)
 4316 CONTINUE
      DO 4317 I = 1,NPATH
      DO 4317 J = 1,21
      EIY1(I,J) = TEN1(I,J) + BB(I)
 4317 CONTINUE
      WRITE(22,378) IK
      DO 4318 I = 1,NPATH
      WRITE(22,377) I
      WRITE(22,275) ((EIY1(I,J)/FA), J= 1,21)
 4318 CONTINUE
      WRITE(22,379)
      WRITE(22,275) ((TEN2(J)/FA),J = 1,101)
      WRITE(22,205)
      H  = H * FACT
      H1 = H1 * FACT
      H2 = H2 * FACT
      IF(NPATH .LE. 1) GO TO 452
      DO 451 J = 1,NPATH
      DO 451 K = 1,2
       FD(J,K) = FD(J,K)/SPAN
 451  CONTINUE
 452  CONTINUE
      HH1 = SPAN1 / (10.0 * SPAN )
      HH2 = SPAN2 / (50.0 * SPAN )
      IF(ISTAGE .EQ. 4) GO TO 471
      IF(ISTAGE .NE. 1) GO TO 460
C
```

```
C     ---------------------------------------------------------------
C         THIS SECTION COMPUTES THE FREQUENCY DETERMINANTS
C         IF ISTAGE = 1
C     ---------------------------------------------------------------

      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,305)
 455   P = H1 * H1
      IF(H1 .GT. H2) GO TO 505
      FR = H1/FACT
      F  = DET(P)
      WRITE(22,275) FR, H1, F
      H1 = H + H1
      GO TO 455
C
```

```
C     ------------------------------------------------------------
C     THIS SECTION COMPUTES THE NATURAL VIBRATION CHARACTERISTICS
C     ------------------------------------------------------------
C
 460  CALL NATFRE(NF)
      IF(IJK .EQ. 0) GO TO 500
      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,370)
      WRITE(22,205)
      DO 461 J = 1, IJK
      FRE   = FREQEN(J) / FACT
      HERTZ = FRE / (2.0 * PI)
      WRITE(22,371) J, FRE, HERTZ
 461  CONTINUE
      WRITE(22,205)
      IF(ISTAGE .EQ. 3) GO TO 505
      ST = SPAN1/(SPAN * 10.0)
      DO 465 J = 1, 11
        SL1(J) = FLOAT(J-1) * ST
 465  CONTINUE
            ST = SPAN2/(SPAN * 50.0)
      DO 470 J = 1, 51
        SL2(J) = SL1(11) + FLOAT(J-1) * ST
 470  CONTINUE
      GO TO 473
 471  DO  472 J = 1, NF
      FREQEN(J) = FREQEN(J) * FACT
 472  CONTINUE
      IJK = NF
 473  DO 495 IJ = 1,IJK
             J1 = IJ
             PP = FREQEN(IJ)
             P  = PP * PP
      CALL SHAPES(P,W1,W2,V1,V2,PHI1,PHI2,U,UC,PSIC,ANUC)
             AMAX = W1(1,1)
         DO 475 I = 1,NPATH
         DO 475 J = 1, 11
           IF(DABS(AMAX) .LT. DABS(W1(I,J))) AMAX = W1(I,J)
           IF(DABS(AMAX) .LT. DABS(V1(I,J))) AMAX = V1(I,J)
           IF(DABS(AMAX) .LT. DABS(PHI1(I,J))) AMAX = PHI1(I,J)
 475     CONTINUE
         DO 480 J = 1, 51
           IF(DABS(AMAX) .LT. DABS(W2(J))) AMAX = W2(J)
           IF(DABS(AMAX) .LT. DABS(V2(J))) AMAX = V2(J)
           IF(DABS(AMAX) .LT. DABS(PHI2(J))) AMAX = PHI2(J)
 480     CONTINUE
         DO 485 I = 1, NPATH
         DO 485 J  = 1, 11
           U(I)    = U(I)/AMAX
           W1(I,J) = W1(I,J)/AMAX
           V1(I,J) = V1(I,J)/AMAX
         PHI1(I,J) = PHI1(I,J)/AMAX
 485     CONTINUE
         DO 490 J = 1,51
            W2(J) = W2(J)/AMAX
```

```
          V2(J) = V2(J)/AMAX
        PHI2(J) = PHI2(J)/AMAX
 490    CONTINUE
        FRE   = PP   / FACT
        HERTZ = FRE / (2.0 * PI)
        CPM   = HERTZ * 60.0
        CALL PLOT(1,   W2,   W1)
        CALL PLOT(2,   V2,   V1)
        CALL PLOT(3, PHI2, PHI1)
     DO 496 J = 1, NPATH
         U(J) = U(J) / AMAX
         WRITE(22,372) J, U(J)
 496 CONTINUE
         PSIC = PSIC / AMAX
         WRITE(22,373) PSIC
         ANUC = ANUC / AMAX
         WRITE(22,374) ANUC
 495 CONTINUE
 500     IF(IJK .LT. NF)   WRITE(22,320) IJK
 505     IF(ISTAGE .EQ. 1) WRITE(22,205)
C
```

```
C     ------------------------------------------------------------
C                                FORMATS
C     ------------------------------------------------------------
C
 100  FORMAT(5I5)
 105  FORMAT(5E14.7)
 110  FORMAT(3A1)
 200  FORMAT(1H1)
 205  FORMAT(//2X,'****************************************',
     1'***************************')
 210  FORMAT(//5X,'NUMBER OF FREQUENCIES REQUIRED = ',I5)
 215  FORMAT(//5X,'FREQUENCY INCREMENTS (RAD/SEC) = ',E14.7)
 220  FORMAT(//5X,'ENDING FREQUENCY (RAD/SEC)     = ',E14.7)
 225  FORMAT(//5X,'RADIUS OF THE ROTOR (INCHES)   = ',E14.7)
 230  FORMAT(//5X,'LENGTH OF INBOARD SEGMENTS (IN)= ',E14.7)
 235  FORMAT(//5X,'LENGTH OF THE BLADE (INCHES)   = ',E14.7)
 236  FORMAT(//5X,'SEMI-CHORD (INCHES)   = ',E14.7)
 240  FORMAT(//5X,'ROTATIONAL SPEED (RPM)         = ',E14.7)
 245  FORMAT(//5X,'NUMBER OF DATA POINTS FOR INBOARD SEGMENTS = ',I5)
 250  FORMAT(//5X,'NUMBER OF DATA POINTS FOR BLADE            = ',I5)
 255  FORMAT(//5X,'PROPERTIES OF THE FIRST LOAD PATH')
 260  FORMAT(//5X,'PROPERTIES OF THE SECOND LOAD PATH')
 265  FORMAT(//5X,'PROPERTIES OF THE THIRD LOAD PATH')
 270  FORMAT(//5X,'DATA POINT LOCATIONS IN INCHES')
 275  FORMAT(4(6X,E14.7))
 280  FORMAT(//5X,'MASS PER UNIT LENGTH (LB-SEC.**2/IN**2)')
 285  FORMAT(//5X,'FLAPWISE BENDING STIFFNESS (LB-IN**2)')
 286  FORMAT(//5X,'AXIAL STIFFNESS(LB) ')
 290  FORMAT(//5X,'PROPERTIES OF THE BLADE')
 295  FORMAT(//5X,'INTERPOLATED VALUES FOR INBOARD SEGMENTS, 21 ',
     1'EQUIDISTANT VALUES')
 300  FORMAT(//5X,'INTERPOLATED VALUES FOR THE BLADE, 101 ',
     1'EQUIDISTANT VALUES')
 305  FORMAT(//1X,'FREQUENCY (RAD/SEC)  NON-DIMENSIONAL FREQUENCY  ',
     1' DETERMINANT')
 310  FORMAT(//5X,'STARTING FREQUENCY (RAD/SEC)  = ',E14.7)
 315  FORMAT(//5X,'NUMBER OF LOAD PATHS          = ',I5)
 320  FORMAT(//5X,'NUMBER OF FREQUENCIES DETECTED WITHIN RANGE',
     1/5X,' = ',I5)
 321  FORMAT(//5X,'Y, Z DISTANCES BETWEEN LOAD PATH NO: ',I2,
     1/5X, ' AND THE BLADE(IN) ARE = ',F6.2,5X,F6.2)
 343  FORMAT(//2X,'AXIAL STIFFNESS OF THE BLADE(LB) IS ')
 347  FORMAT(//5X,'EXECUTION STAGE REQUIRED IS  = ',I5)
 348  FORMAT(//5X,'PLOT INDEX IS = ',I5)
 350  FORMAT(//5X,'CHORDWISE BENDING STIFFNESS(LB-IN**2)')
 352  FORMAT(//5X,'TORSIONAL STIFFNESS(LB-IN**2)')
 354  FORMAT(//5X,'DISTANCE BETWEEN MASS AND ELASTIC AXIS(IN) =')
 356  FORMAT(//5X,'TWIST OF THE BLADE INCLUDING THE COLLECTIVE',
     1/5X,'(DEGREES) =')
 358  FORMAT(//5X,'MASS MOMENT OF INERTIA ABOUT THE CHORD(LB-SEC**2)')
 360  FORMAT(//5X,'MASS MOMENT OF INERTIA ABOUT AN AXIS ',
     1/5X,'PERPENDICULAR TO THE CHORD THROUGH THE CENTER OF ',
     1/5X,'GRAVITY(LB-SEC**2)')
 370  FORMAT(/5X,'THE NATURAL FREQUENCIES OF THE SYSTEM ARE:',
     1/17X,'  RAD / SEC: ', '      HERTZ: ')
 371  FORMAT(5X, I5, 5X, F11.4, 5X, F11.4)
```

```
 372  FORMAT(//5X,'AXIAL  DISPLACEMENT  OF  LOAD  PATH # ',I3,2X,
     1/5X, 'AT THE CLEVIS = ',E11.4)
 373  FORMAT(//5X,'FLAPWISE  BENDING SLOPE AT THE CLEVIS = ',E11.4)
 374  FORMAT(//5X,'CHORDWISE BENDING SLOPE AT THE CLEVIS = ',E11.4)
 375  FORMAT(//5X,'NUMBER OF TENSION ITERATIONS = ',I5)
 376  FORMAT(//5X,'THE FOLLOWING ARE THE STARTING TENSIONS')
 377  FORMAT(//5X,'THE TENSION COEFFICIENTS IN LOAD'
     1/5X,'PATH #',I5,' ARE')
 378  FORMAT(//5X,'THE TENSIONS AFTER ITERATION #',I5, ' ARE')
 379  FORMAT(//5X,'THE TENSIONS IN THE BLADE ARE')
 380  FORMAT(//5X,'STARTING TENSIONS IN THE LOAD PATHS AT THE'
     1         /5X,'CLEVIS ARE')
      STOP
      END
C
```

```
C     ***************************************************************
                                  FUNCTION DET(P)
C     ***************************************************************
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION TF1(3,12,12), TF2(12,12), A(6,6), B(6,6),
     1 C(6,6), D(6,6), E(6,6),  R(6,6), FD(3,2)
      COMMON/X4/NPATH,ICPL
      COMMON/X9/FD
      COMMON/X10/TF1, TF2
      COMMON/XS2/DETER
      CALL TRAMAT(P)
      DO 10 I = 1, 6
      DO 10 J = 1, 6
         A(I,J) = TF2(I+6,   J)
         B(I,J) = TF2(I+6, J+6)
         R(I,J) = 0.0
 10   CONTINUE
      DO   20  I = 1, NPATH
         DO 15 J = 1, 6
         DO 15 K = 1, 6
          C(J,K) = TF1(I, J, K+6)
          D(J,K) = TF1(I, J+6, K+6)
 15      CONTINUE
      CALL SOLUTN(C, 6, -1, 6)
      DO 16 J = 1, 6
       C(J,4) = C(J,4) - FD(I,2) * C(J,1)
       C(J,5) = C(J,5) - FD(I,1) * C(J,1)
 16   CONTINUE
 17   CALL MATMUL(6, 6, 6, D, C, E)
      DO 18  J = 1,6
        E(1,J) = E(1,J) - FD(I,2) * E(4,J) + FD(I,1) * E(5,J)
        E(2,J) = E(2,J) - FD(I,1) * E(6,J)
        E(3,J) = E(3,J) + FD(I,2) * E(6,J)
 18   CONTINUE
         DO 20 J = 1, 6
         DO 20 K = 1, 6
          R(J,K) = R(J,K) + E(J,K)
 20   CONTINUE
      CALL MATMUL(6, 6, 6, B, R, D)
      DO 25 J = 1, 6
      DO 25 K = 1, 6
         B(J,K) = D(J,K) + A(J,K)
 25   CONTINUE
      CALL SOLUTN(B, 6, -1, 6)
      DET = DETER
      RETURN
      END
C
```

```
C     *****************************************************************
                 SUBROUTINE INTPOL(N, A, H)
C     *****************************************************************
C
C     -----------------------------------------------------------------
C      THIS SUBROUTINE INTERPOLATES FOR THE REQUIRED VALUES
C     -----------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION A(101), STA(101), TABLE(101,1), B(101)
      COMMON/X3/STA, NS
      NN = N - 1
      A(N) = A(NS)
      NM1 = NS -1
      DO 20 I = 1, NM1
 20   TABLE(I,1) = (A(I+1) - A(I))/(STA(I+1) - STA(I))
      XARG = H
      DO 35 I = 2, NN
         DO 25 J = 1, NS
            IF(J .EQ. NS .OR. XARG .LE. STA(J)) GO TO 30
 25      CONTINUE
 30      MAX = J
         IF(MAX .LE. 2) MAX = 2
         ISUB = MAX - 1
         YEST = TABLE(ISUB,1)
         B(I) = YEST * (XARG - STA(ISUB)) + A(ISUB)
 35   XARG = XARG + H
      DO 40 J = 2, NN
 40   A(J) = B(J)
      RETURN
      END
C
```

```
C     **********************************************************************
            SUBROUTINE MATMUL(L, M, N, A, B, C)
C     **********************************************************************
C
C     ----------------------------------------------------------------------
C                          MATRIX MULTIPLICATION
C     ----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION A(6,6), B(6,6), C(6,6)
      DO 10 I = 1, L
      DO 10 J = 1, N
       C(I,J) = 0.0
      DO 10 K = 1, M
       C(I,J) = C(I,J) + A(I,K) * B(K,J)
 10   CONTINUE
      RETURN
      END
C
```

```
C     ******************************************************************
                     SUBROUTINE MROOT(SDT, H, FRE, IJ, ICOUNT)
C     ******************************************************************
C
C     ------------------------------------------------------------------
C   THIS SUBROUTINE CALCULATES THE MISSING FREQUENCIES, IF ANY
C     ------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION SDT(2,2000), FREMIS(10), FRE(10)
      COMMON/X2/FACT
      INDEX = 0
      N = ICOUNT - 2
      IJ = 0
      DO 10 I = 1, N
         A = SDT(2,I)
         B = SDT(2,I+1)
         C = SDT(2,I+2)
         D = A * B
         IF(D .LT. 0.0) GO TO 10
         D = B * C
         IF(D .LT. 0.0) GO TO 10
         D = DABS(A) - DABS(B)
         IF(D .LT. 0.0) GO TO 10
         D = DABS(C) - DABS(B)
         IF(D .LT. 0.0) GO TO 10
         INDEX = INDEX + 1
         IF(INDEX .GT. 5) GO TO 15
         FREMIS(INDEX) = SDT(1,I)
 10   CONTINUE
      IF(INDEX .EQ. 0) GO TO 90
 15   CONTINUE
      DO 60 I = 1, INDEX
         NS = 1
 20      HH = H/(FLOAT(NS) * 10.0)
         PP = FREMIS(INDEX)
         P  = PP * PP
         F  = DET(P)
         F  = DSIGN(1.0D+00,F)
         A  = FREMIS(INDEX) + 2.0 * H
 30      PP = PP + HH
              IF(PP .GT. A) GO TO 50
         P  = PP * PP
         G  = DET(P)
         G  = DSIGN(1.0D+00,G)
              IF(F*G .GT. 0.0) GO TO 40
         IJ = IJ + 1
         P  = (PP-HH) * (PP-HH)
         C  = DET(P)
         P  = PP * PP
         D  = DET(P)
         FRE(IJ) = PP - D * HH/(D - C)
         IF((IJ/2*2-IJ) .EQ. 0) GO TO 60
 40      F  = G
              GO TO 30
 50      NS = NS * 10
```

```
            IF(NS .GT. 100) GO TO 70
            GO TO 20
 60   CONTINUE
 70   WRITE(22,100)
      WRITE(22,110)
      DO 80 I = 1, INDEX
            A = FREMIS(I)/FACT
            B = A + 2.0 * H/FACT
         WRITE(22,120) A, B
 80   CONTINUE
 90   CONTINUE
 100  FORMAT(1H1)
 110  FORMAT(10X,'CHECK FOR TWO FREQUENCIES BETWEEN EACH OF THE
     1',/10X,'FOLLOWING SETS. OTHERWISE, THEY ARE MISSING')
 120  FORMAT(2(10X,E14.7))
      RETURN
      END
C
```

```
C     ******************************************************************
                                SUBROUTINE NATFRE(N)
C     ******************************************************************
C
C     ------------------------------------------------------------------
C  THIS SUBROUTINE SCANS THE FREQUENCY DETERMINANT WITH RESPECT TO
C  THE FREQUENCY UNTIL THE SPECIFIED NUMBER OF SIGN CHANGES ARE
C  DETECTED, STARTING FROM ZERO FREQUENCY. USES THE "DET" ROUTINE
C     ------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION FREQEN(10), JKL(10), SDT(2,2000), FRE(10), STO(20)
      COMMON/X1/FREQEN, H, H1, H2, IJK
      IJK    = 0
      IJ     = 0
      ICOUNT = 1
      DO 5 J = 1, N
 5    JKL(J) = 0
          PP = H1
           P = PP * PP
           F = DET(P)
      SDT(1,ICOUNT) = PP
      SDT(2,ICOUNT) = F
      IF(DABS(F) .GT. 0.0001) GO TO 10
           IJK      = IJK + 1
           JKL(IJK) = 1
        FREQEN(IJK) = PP
                 PP = PP + H
                 P  = PP * PP
                 F  = DET(P)
             ICOUNT = ICOUNT + 1
      SDT(1,ICOUNT) = PP
      SDT(2,ICOUNT) = F
 10   F  = DSIGN(1.0D+00,F)
 15   PP = PP + H
           IF(PP .GT. H2) GO TO 55
      P  = PP * PP
      G = DET(P)
      ICOUNT = ICOUNT + 1
      SDT(1,ICOUNT) = PP
      SDT(2,ICOUNT) = G
      IF(DABS(G) .GT. 0.0001) GO TO 20
      IJK = IJK + 1
      JKL(IJK) = 1
      FREQEN(IJK) = PP
      IF(IJK .EQ. N) GO TO 55
      PP = PP + H
      P = PP * PP
      F = DET(P)
      ICOUNT = ICOUNT + 1
      SDT(1,ICOUNT) = PP
      SDT(2,ICOUNT) = F
      F = DSIGN(1.0D+00,F)
      GO TO 15
 20   G = DSIGN(1.0D+00,G)
      IF(F*G .GT. 0.0) GO TO 25
```

```
      IJK = IJK + 1
      FREQEN(IJK) = PP - H
      IF(IJK .EQ. N) GO TO 55
25    F = G
      GO TO 15
55    IF(IJK .EQ. 0) GO TO 65
      HS = H/10.0
      DO 50 J = 1, IJK
         IF(JKL(J) .EQ. 1) GO TO 50
         PP = FREQEN(J)
         P = PP * PP
         F = DET(P)
         F = DSIGN(1.0D+00,F)
35       PP = PP + HS
         P = PP * PP
         G = DET(P)
         IF(DABS(G) .GT. 0.0001) GO TO 40
         JKL(J) = 1
         FREQEN(J) = PP
         GO TO 50
40       G = DSIGN(1.0D+00,G)
         IF(F*G .GT. 0.0) GO TO 45
         FREQEN(J) = PP - HS
         GO TO 50
45       F = G
         GO TO 35
50    CONTINUE
      DO 60 J = 1, IJK
         IF(JKL(J) .EQ. 1) GO TO 60
         PP = FREQEN(J)
         P = PP * PP
         F = DET(P)
         PP = PP + HS
         P = PP * PP
         G = DET(P)
         DIFF = G - F
         FREQEN(J) = PP - G*HS/DIFF
60    CONTINUE
65    CONTINUE
      CALL MROOT(SDT, H, FRE, IJ, ICOUNT)
      IF(IJ .EQ. 0) GO TO 100
      N2 = IJK + IJ
      IF(IJK .EQ. 0) GO TO 75
      DO 70 I = 1, IJK
70    STO(I) = FREQEN(I)
75    N1 = IJK + 1
      DO 80 I = N1, N2
80    STO(I) = FREQEN(I-IJK)
      DO 90 I = 2, N2
      DO 90 J = I, N2
         IF(STO(I-1) - STO(J)) 90, 90, 85
85       STORE = STO(J)
         STO(J) = STO(I-1)
         STO(I-1) = STORE
90    CONTINUE
      IJK = N2
```

```
      IF(N2 .GT. N) IJK = N
      DO 95 I = 1, IJK
 95   FREQEN(I) = STO(I)
 100  CONTINUE
      RETURN
      END
C
```

```
C     ******************************************************************
                        SUBROUTINE PLOT(NN, C, B)
C     ******************************************************************
C
C     ------------------------------------------------------------------
C        PRINTS AND PLOTS NATURAL VIBRATION CHARACTERISTICS
C     ------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      REAL*8 LINE(61)
      DIMENSION C(51), B(3,11), SL1(11), SL2(51), A(61), SL(61)
      COMMON/X4/NPATH,ICPL
      COMMON/X6/SL1,SL2,CPM,FRE,HERTZ,BLANK,DOT,STAR,J1,IPLOT
      WRITE(22,10)
      WRITE(22,20)
      WRITE(22,30) J1, FRE, HERTZ, CPM
      WRITE(22,20)
      IF(NN .EQ. 1)WRITE(22,39)
      IF(NN .EQ. 2)WRITE(22,40)
      IF(NN .EQ. 3)WRITE(22,41)
      DO 50 J = 1, 11
         SL(J) = SL1(J)
         A(J) = B(1,J)
 50   CONTINUE
      DO 60 J = 2, 51
         SL(J+10) = SL2(J)
         A(J+10) = C(J)
 60   CONTINUE
      WRITE(22,70)
      DO 80 J = 1, 11
      WRITE(22,90) SL(J),A(J),SL(J+21),A(J+21),SL(J+42),A(J+42)
 80   CONTINUE
      DO 81 J = 12,19
      WRITE(22,90) SL(J),A(J),SL(J+21),A(J+21),SL(J+42),A(J+42)
 81   CONTINUE
      WRITE(22,90) SL(20), A(20), SL(41), A(41)
      WRITE(22,90) SL(21), A(21), SL(42), A(42)
      WRITE(22,20)
      IF(NPATH .LE. 1) GO TO 140
      WRITE(22,10)
      WRITE(22,20)
      DO 130 I = 2, NPATH
       IF(NN .EQ. 1)  WRITE(22,100) I
       IF(NN .EQ. 2)  WRITE(22,101) I
       IF(NN .EQ. 3)  WRITE(22,102) I
         WRITE(22,70)
         DO 120 J = 1, 3
      WRITE(22,90)  SL1(J),B(I,J),SL1(J+4),B(I,J+4),SL1(J+8),
     1   B(I,J+8)
 120  CONTINUE
      WRITE(22,90) SL1(4), B(I,4), SL1(8), B(I,8)
 130  CONTINUE
 140  WRITE(22,20)
      WRITE(22,10)
      IF(IPLOT .EQ. 0) GO TO 180
      DO 150 J = 1, 61
```

```
         LINE(J) = DOT
 150  CONTINUE
      J = 30.0 * (A(1) + 1.0) + 1.5
      LINE(J) = STAR
      WRITE(22,110) (LINE(J),J = 1,61)
      DO 160 J = 1, 61
         LINE(J) = BLANK
 160  CONTINUE
      LINE(31) = DOT
      DO 170 JJ = 2, 61
         J = 30.0 * (A(JJ) + 1.0) + 1.5
         IF(J .GT. 61) GO TO 170
         LINE(J) = STAR
         WRITE(22,190) (LINE(JV), JV = 1,61)
         LINE(J) = BLANK
         LINE(31) = DOT
 170  CONTINUE
 180  CONTINUE
 10   FORMAT(1H1)
 20   FORMAT(/2X,'**************************************************',
     1'********************')
 30   FORMAT(//5X,'MODE NUMBER = ',I2,8X,'FREQ. RAD/SEC = ',F10.4,8X,
     1/5X,'FREQ. HERTZ = ',F10.4,8X,'FREQ. CYCLES/MINUTE = ',F10.4)
 39   FORMAT(//15X,'    FLAPWISE DEFLECTION/SEMI-CHORD  ')
 40   FORMAT(//15X,'   CHORDWISE DEFLECTION/SEMI-CHORD  ')
 41   FORMAT(//15X,'   TORSIONAL DEFLECTION (RAD)')
 70   FORMAT(//4X,'STA X/L',3X,'DEFLN',9X,'STA X/L',3X,'DEFLN',9X,
     1'STA X/L',3X,'DEFLN')
 90   FORMAT(//3(2X,F8.4,3X,E11.4))
 100  FORMAT(//5X,'FLAPWISE DEFLECTION/SEMI-CHORD    LOAD PATH # ',I5)
 101  FORMAT(//5X,'CHORDWISE DEFLECTION/SEMI-CHORD   LOAD PATH # ',I5)
 102  FORMAT(//5X,'TORSIONAL DEFLECTION (RAD)        LOAD PATH # ',I5)
 110  FORMAT(//4X,61A1)
 190  FORMAT(4X,61A1)
 1101 RETURN
      END
C
```

```
C     *****************************************************************
                           FUNCTION RUNGE(Y, F, J, M, HH)
C     *****************************************************************
C
C     -----------------------------------------------------------------
C                        FOURTH ORDER RUNGE-KUTTA METHOD
C     -----------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      INTEGER RUNGE
      DIMENSION Y(12), F(12), PHI(12), SAVEY(12)
      NN = 12
      M = M + 1
      GO TO(5, 10, 20, 30, 40), M
 5    RUNGE = 1
      RETURN
 10   DO 15 JJ = 1, NN
         SAVEY(JJ) = Y(JJ)
         PHI(JJ) = F(JJ)
         Y(JJ) = SAVEY(JJ) + HH * F(JJ) / 2.0
 15   CONTINUE
      J = J + 1
      RUNGE = 1
      RETURN
 20   DO 25 JJ = 1, NN
         PHI(JJ) = PHI(JJ) + 2.0 * F(JJ)
         Y(JJ) = SAVEY(JJ) + HH * F(JJ) / 2.0
 25   CONTINUE
      RUNGE = 1
      RETURN
 30   DO 35 JJ = 1, NN
         PHI(JJ) = PHI(JJ) + 2.0 * F(JJ)
         Y(JJ) = SAVEY(JJ) + HH * F(JJ)
 35   CONTINUE
      J=J+1
      RUNGE = 1
      RETURN
 40   DO 45 JJ = 1, NN
 45   Y(JJ) = SAVEY(JJ) + (PHI(JJ) + F(JJ))* HH / 6.0
      M = 0
      RUNGE = 0
      RETURN
      END
C
```

```
C     *****************************************************************
      SUBROUTINE SHAPES(P,W1,W2,V1,V2,PHI1,PHI2,U,UC,PSIC,ANUC)
C     *****************************************************************
C
C     ---------------------------------------------------------------
C         THIS SUBROUTINE COMPUTES THE NATURAL MODE SHAPES
C     ---------------------------------------------------------------
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION W1(3,11),W2(51),V1(3,11), V2(51),
     1 PHI1(3,11), PHI2(51),U(3),TF1(3,12,12),
     1TF2(12,12),TT1(3,11,3,12),TT2(51,3,12),
     1A(6,6),B(6,6),C(6,6),D(6,6),E(6,6),
     1FD(3,2),R(6,6),BB(12),TEM(12,12) , G(5,5)
      COMMON/X4/NPATH,ICPL
      COMMON/X8/BB,IND
      COMMON/X9/FD
      COMMON/X10/TF1, TF2
      COMMON/X12/CON
      COMMON/XS3/TT1, TT2
      CALL TRAMAT(P)
      DO  5 I = 1,12
      DO  5 J = 1,12
      TEM(I,J) = TF2(I,J)
 5    CONTINUE
      DO  10 I = 1,6
      DO  10 J = 1,6
      A(I,J) = TF2(I+6,J)
      B(I,J) = TF2(I+6,J+6)
      R(I,J) = 0.0
 10   CONTINUE
      DO 20 I = 1, NPATH
      DO  15 J = 1,6
      DO  15 K = 1,6
      C(J,K)   = TF1(I,J,K+6)
      D(J,K)   = TF1(I,J+6,K+6)
 15   CONTINUE
      CALL SOLUTN(C,6,-1,6)
      DO 16 J = 1,6
      C(J,4) = C(J,4) - FD(I,2) * C(J,1)
      C(J,5) = C(J,5) - FD(I,1) * C(J,1)
 16   CONTINUE
 17   CALL MATMUL(6,6,6,D,C,E)
      DO 18 J = 1,6
      E(1,J) = E(1,J) - FD(I,2) * E(4,J)+ FD(I,1) * E(5,J)
      E(2,J) = E(2,J) - FD(I,1) * E(6,J)
      E(3,J) = E(3,J) + FD(I,2) * E(6,J)
 18   CONTINUE
      DO 20 J = 1,6
      DO 20 K = 1,6
      R(J,K) = R(J,K) + E(J,K)
 20   CONTINUE
      CALL MATMUL(6,6,6,B,R,D)
       DO  25 J = 1,6
       DO  25 K = 1,6
       E(J,K) = D(J,K) + A(J,K)
```

```
25     CONTINUE
      CALL SOLUTN(TEM,12,-1,12)
        DO  30 J = 1,6
        DO  30 K = 1,6
        D(J,K) = TEM(J,K)
30      CONTINUE
      CALL MATMUL(6,6,6,E,D,A)
      WRITE(24,111) ((A(I,J),J = 1,6),I = 1,6)
111   FORMAT(/(6(E11.4,1X)))
      IF(ICPL .NE. 0) GO TO 310
      C(1,1) = 0.0
      C(2,1) = 1.0
      DO 32 K = 1,4
      DO 31 J = 1,4
      G(J,K) = A(J,K+2)
31    CONTINUE
      G(K,5) = -A(K,2)
32    CONTINUE
      CALL SOLUTN(G,4,1,5)
      DO 33 J = 3,6
      C(J,1) = BB(J-2)
33    CONTINUE
      GO TO 375
310   DO 36 K = 1,5
      KK = K
      IF(K .GE. 2) KK = K + 1
      DO 35 J = 1,5
      D(J,K) = A(J,KK)
35    CONTINUE
      D(K,6) = -A(K,2)
36    CONTINUE
      CALL SOLUTN(D,5,1,6)
      C(1,1) = BB(1)
      C(2,1) = 1.0
      DO 37 J = 3,6
      C(J,1) = BB(J-1)
37    CONTINUE
375   CONTINUE
      DO 376 I = 1,6
      A(I,1) = 0.0
      B(I,1) = 0.0
      DO 376 K = 1,6
      A(I,1) = A(I,1) + TEM(I,K) * C(K,1)
      B(I,1) = B(I,1) + TEM(I+6,K) * C(K,1)
376   CONTINUE
      DO 39 I = 1,51
      DO 39 J = 1,3
      STO = 0.0
      DO 38 K = 1,6
      STO = STO + TT2(I,J,K) * A(K,1) + TT2(I,J,K+6) * B(K,1)
38    CONTINUE
      PHI2(I) = STO
      IF(J .EQ. 1) W2(I) = STO * CON
      IF(J .EQ. 2) V2(I) = STO * CON
39    CONTINUE
      DO 48 J = 2,6
```

```
      C(J,1) = A(J,1)
48    CONTINUE
      DO 60 I = 1,NPATH
      C(1,1)=A(1,1)-FD(I,2)*A(4,1)-FD(I,1)*A(5,1)
      DO 50 J = 1,6
      DO 50 K = 1,6
      B(J,K) = TF1(I,J,K+6)
50    CONTINUE
      CALL SOLUTN(B,6,-1,6)
      CALL MATMUL(6,6,1,B,C,D)
      DO 55 J = 1,11
      W1(I,J) = 0.0
      V1(I,J) = 0.0
      PHI1(I,J) = 0.0
      DO 55 K = 1,6
      W1(I,J) = W1(I,J) + ( TT1(I,J,1,K+6) * D(K,1) ) * CON
      V1(I,J) = V1(I,J) + ( TT1(I,J,2,K+6) * D(K,1) ) * CON
      PHI1(I,J) = PHI1(I,J) + TT1(I,J,3,K+6) * D(K,1)
55    CONTINUE
      U(I) = C(1,1)
60    CONTINUE
      UC = A(1,1)
      PSIC = A(4,1)
      ANUC = A(5,1)
      RETURN
      END
C
```

```
C     ****************************************************************
                           SUBROUTINE SOLUTN(A, N, INDIC, NRC)
C     ****************************************************************
C
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION A(NRC,NRC),X(12),IROW(12),JCOL(12),JORD(12),Y(12)
      COMMON/XS2/DETER
      COMMON/X8/X, IND
      IND = 0
      MAX = N
      IF(INDIC .GE. 0) MAX = N + 1
      DETER = 1.0
      DO 80 K = 1, N
         KM1 = K - 1
         PIVOT = 0.0
         DO 60 I = 1, N
         DO 60 J = 1, N
            IF(K .EQ. 1) GO TO 55
            DO 50 ISCAN = 1, KM1
            DO 50 JSCAN = 1, KM1
               IF(I .EQ. IROW(ISCAN)) GO TO 60
               IF(J .EQ. JCOL(JSCAN)) GO TO 60
 50         CONTINUE
 55         IF(DABS(A(I,J)) .LE. DABS(PIVOT)) GO TO 60
            PIVOT = A(I,J)
            IROW(K) = I
            JCOL(K) = J
 60       CONTINUE
          IF(DABS(PIVOT) .GT. 0.1E-20) GO TO 65
          DETER = 0.0
          IND = 1
          RETURN
 65       IROWK = IROW(K)
          JCOLK = JCOL(K)
          DETER = DETER * PIVOT
          DO 70 J = 1, MAX
 70       A(IROWK,J) = A(IROWK,J)/PIVOT
          A(IROWK,JCOLK) = 1.0/PIVOT
          DO 80 I = 1, N
             AIJCK = A(I,JCOLK)
             IF(I .EQ. IROWK) GO TO 80
             A(I,JCOLK) = -AIJCK/PIVOT
             DO 75 J = 1, MAX
 75          IF(J .NE. JCOLK) A(I,J) = A(I,J) - AIJCK * A(IROWK,J)
 80   CONTINUE
      DO 85 I = 1, N
         IROWI = IROW(I)
         JCOLI = JCOL(I)
         JORD(IROWI) = JCOLI
         IF(INDIC .GE. 0) X(JCOLI) = A(IROWI,MAX)
 85   CONTINUE
      INTCH = 0
      NM1 = N-1
      DO 90 I = 1, NM1
         IP1 = I + 1
         DO 90 J = IP1, N
```

```
          IF(JORD(J) .GE. JORD(I)) GO TO 90
          JTEMP = JORD(J)
          JORD(J) = JORD(I)
          JORD(I) = JTEMP
          INTCH = INTCH + 1
 90    CONTINUE
       IF(INTCH/2*2 .NE. INTCH) DETER = -DETER
       IF(INDIC .LE. 0) GO TO 94
       RETURN
 94    DO 100 J = 1, N
          DO 95 I = 1, N
             IROWI = IROW(I)
             JCOLI = JCOL(I)
             Y(JCOLI) = A(IROWI,J)
 95       CONTINUE
          DO 100 I = 1, N
             A(I,J) = Y(I)
 100      CONTINUE
          DO 110 I = 1, N
             DO 105 J = 1, N
                IROWJ = IROW(J)
                JCOLJ = JCOL(J)
                Y(IROWJ) = A(I,JCOLJ)
 105         CONTINUE
           DO 110 J = 1, N
               A(I,J) = Y(J)
 110       CONTINUE
          RETURN
          END
C
```

```
C     ******************************************************************
                                  SUBROUTINE TRAMAT(P)
C     ******************************************************************
C
C     ------------------------------------------------------------------
C        THIS SUBROUTINE COMPUTES THE TRANSFER MATRIX
C     ------------------------------------------------------------------
C
       IMPLICIT REAL*8 (A - H, O - Z)
       INTEGER RUNGE
       DIMENSION TF1(3,12,12), TF2(12,12), TT1(3,11,3,12),
      1TT2(51,3,12),T(12),V(12), TEN1(3,21), TEN2(101), EA1(3,21),
      1EA2(101),D11(3,21),D21(101),D12(3,21),D22(101),D13(3,21),
      1D23(101),D14(3,21),D24(101),D15(3,21),D25(101),D16(3,21),
      1D26(101),D17(3,21),D27(101),D18(3,21),D28(101),D19(3,21),
      1D29(101),D191(3,21),D291(101),D192(3,21),D292(101)
       COMMON/X4/NPATH,ICPL
       COMMON/X5/TEN1,TEN2,EA1,EA2,D11,D21,D12,D22,D13,
      1D23,D14,D24,D15,D25,D16,D26,D17,D27,D18,D28,D19,
      1D29,D191,D291,D192,D292,OMEGAN
       COMMON/X7/HH1,HH2
       COMMON/X10/TF1, TF2
       COMMON/XS3/TT1, TT2
       HH = HH1
       PP = P + OMEGAN
       DO 35 I = 1, NPATH
       DO 35 J = 1, 12
          DO 10 K = 1, 12
             V(K) = 0.0
 10       CONTINUE
          V(J) = 1.0
          M = 0
          NJ = 1
          DO 25 L = 1, 10
 15          K = RUNGE(V, T, NJ, M, HH)
             IF(K .NE. 1) GO TO 20
             T(1) = EA1(I,NJ)*V(12)
             T(2) = V(4)
             T(3) = V(5)
             T(4) = D19(I,NJ) * V(8) + D191(I,NJ) * V(9)
             T(5) = D192(I,NJ) * V(8) - D19(I,NJ) * V(9)
             T(6) = D11(I,NJ) * V(7)
             T(7) = -P * D13(I,NJ) * V(2) + PP * D14(I,NJ) * V(3)
     1              + D15(I,NJ) * V(4) - D16(I,NJ) * V(5)
     1              + (D17(I,NJ) - P * D18(I,NJ) ) * V(6)
             T(8) = TEN1(I,NJ) * V(5) - D16(I,NJ) * V(6) - V(10)
             T(9) = -TEN1(I,NJ) * V(4) - D15(I,NJ) * V(6) + V(11)
             T(10)= -PP * D12(I,NJ) * V(3) + PP * D14(I,NJ) * V(6)
             T(11)= -P * D12(I,NJ) * V(2) - P * D13(I,NJ) * V(6)
             T(12)= -PP * D12(I,NJ) * V(1)
             GO TO 15
 20    TT1(I,L+1,1,J) = V(2)
       TT1(I,L+1,2,J) = V(3)
       TT1(I,L+1,3,J) = V(6)
 25    CONTINUE
       DO 30 IJ = 1,12
```

```
      TF1(I,IJ,J) = V(IJ)
30    CONTINUE
35    CONTINUE
      HH = HH2
      DO 65 J = 1, 12
         DO 40 K = 1, 12
            V(K) = 0.0
40       CONTINUE
         V(J) = 1.0
         M = 0
         NJ = 1
         DO 55 L = 1, 50
45       K = RUNGE(V, T, NJ, M, HH)
         IF(K .NE. 1) GO TO 50
            T(1) = EA2(NJ)*V(12)
            T(2) = V(4)
            T(3) = V(5)
            T(4) = D29(NJ) * V(8) + D291(NJ) * V(9)
            T(5) = D292(NJ) * V(8) - D29(NJ) * V(9)
            T(6) = D21(NJ) * V(7)
            T(7) = -P * D23(NJ) * V(2) + PP * D24(NJ) * V(3)
     1             + D25(NJ) * V(4) - D26(NJ) * V(5)
     1             + (D27(NJ) - P * D28(NJ) ) * V(6)
            T(8) = TEN2(NJ) * V(5) - D26(NJ) * V(6) - V(10)
            T(9) = -TEN2(NJ) * V(4) - D25(NJ) * V(6) + V(11)
            T(10)= -PP * D22(NJ) * V(3) + PP * D24(NJ) * V(6)
            T(11)= -P * D22(NJ) * V(2) - P * D23(NJ) * V(6)
            T(12)= -PP * D22(NJ) * V(1)
         GO TO 45
50       TT2(L+1,1,J) = V(2)
         TT2(L+1,2,J) = V(3)
         TT2(L+1,3,J) = V(6)
55    CONTINUE
      DO 60 IJ = 1,12
      TF2(IJ,J) = V(IJ)
60    CONTINUE
65    CONTINUE
      DO 75 I = 1, NPATH
      DO 70 J = 1,3
      DO 70 K = 1,12
      TT1(I,1,J,K) = 0.0
70    CONTINUE
      TT1(I,1,1,2) = 1.0
      TT1(I,1,2,3) = 1.0
      TT1(I,1,3,6) = 1.0
75    CONTINUE
      DO 80 J = 1,3
      DO 80 K = 1,12
      TT2(1,J,K) = 0.0
80    CONTINUE
      TT2(1,1,2) = 1.0
      TT2(1,2,3) = 1.0
      TT2(1,3,6) = 1.0
      RETURN
      END
```

APPENDIX B

APPENDIX B

USER'S INSTRUCTIONS

This program computes the natural frequencies and the associated mode shapes of a nonuniform pretwisted multiple load path rotating rotor blade with coupled flapwise bending, chordwise bending and torsional degrees of freedom with differential axial displacements in the load paths.

I DATA CARD

* NPATH, ISTAGE, IPLOT, NS1, NS2

* FORMAT(5 I 5)

NPATH: Number of load paths, maximum = 3.

ISTAGE: Program performs four functions:

1 - computes the values for frequency determinants only.

2 - computes natural frequencies and mode shapes.

3 - computes natural frequencies only.

4 - computes mode shapes corresponding to given natural frequencies.

IPLOT: 0 - no mode shape plots.

1 - plots mode shapes.

It is preferable to provide data such that the variation of the properties of the blade are linear or uniform between any two stations. For a uniform blade, it is enough to provide the data at the axis of rotation and the clevis for the load paths and clevis and the tip for the blade.

Note: Load paths - First station should correspond to the axis of rotation and the last station should correspond to the clevis.

Blade - First station should correspond to the clevis and the last station should correspond to the tip of the blade.

II DATA CARD SET

* SPAN1, SPAN2, SCH, OMEGA

* FORMAT(5 E 14.7)

SPAN1: span of the load paths (inches).

SPAN2: span of the blade (inches).
Note: SPAN1 + SPAN2 = Radius of the rotor.

SCH: semi-chord of the blade (inches).

OMEGA: rotational speed, (rpm).

The following data should be provided in the given order. The format is (5 E 14.7) unless otherwise specified.

1. STA1: station locations for load paths, (inches).

2. STA2: station locations for blade, (inches).

The following data corresponds to a single load path system:

3. MASS1: mass/unit length (lb-sq.sec/sq.in).

4. EIY1: flapwise bending stiffness (lb-sq.in).

5. EIZ1: chordwise bending stiffness (lb-sq.in).

6. GJ1: torsional stiffness (lb-sq.in).

7. E1: distance between mass and elastic axis (inches).

8. BETA1: twist of the load path including the collective (degrees).

9. KM1S1: mass moment of inertia about the chordwise axis. (lb-sq.sec).

10. KM2S1: mass moment of inertia about an axis perpendicular to the chordwise axis through the c.g. (lb-sq.sec).

11. EA1: axial stiffness (lb).

12. REPEAT THE DATA 3 TO 11 FOR LOAD PATHS 2 AND 3.

The following data corresponds to the blade:

13. MASS2: Mass/unit length (lb-sq.sec/sq.in).

14. EIY2: flapwise bending stiffness (lb-sq.in).

15. EIZ2: chordwise bending stiffness (lb-sq.in).

16. GJ2: torsional stiffness (lb-sq.in).

17. E2: distance between mass and elastic axis (inches).

18. BETA2: twist of the blade including the collective (degrees).

19. KM1S2: mass moment of inertia about the chordwise axis. (lb-sq.sec).

20. KM2S2: mass moment of inertia about an axis perpendicular to the chordwise axis through the c.g. (lb-sq.sec).

21. EA2: axial stiffness (lb).

III DATA CARD

* H1, H, H2

* FORMAT(3 E 14.7)

H1: starting frequency (rad/sec).

H: frequency increment (rad/sec).

H2: ending frequency (rad/sec).

CASE 1: ISTAGE = 1

In this case, only frequency determinants are computed for various frequencies. It starts with frequency H1, increments by H until it reaches the value H2.

CASE 2: ISTAGE = 2

In this case, the natural frequencies and mode shapes are computed by the frequency scanning technique. Values of frequency determinant are scanned starting from the value H1 at steps of H until the required sign changes are detected or the value H2 is reached. So the required number of frequencies or the number of frequencies that lie between H1 and H2 whichever is less, are computed. If two frequencies are closer than increment H, then there is a possibility of missing these frequencies. In such cases the frequencies are detected by the fact that if any three consecutive determinants have the same sign and the absolute value of the middle determinant is the smallest of the three then there are two frequencies in that range. In this case smaller increments are taken to bracket the missing frequencies. However there exists a remote possibility of missing these roots also. In such a case a closer scanning of the frequency determinants is required.

IV DATA CARD

* FD(I,J)

* FORMAT(5 E 14.7)

This card is required if NPATH is greater than 1 and gives the location of the j-th load path with respect to the clevis (inches).

FD(1,J) = h_{y_j}

FD(2,J) = h_{z_j}

V DATA CARD

* ITR - number of tension iterations.

* FORMAT(I 5)

This card is required if NPATH > 1. ITR = 2 will be enough.

VI DATA CARD

* NF

* FORMAT(I 5)

This card is required if ISTAGE is not equal to 1.

NF = number of frequencies, maximum = 10

VII DATA CARD

* FREQEN

* FORMAT(5 E 14.7)

This card is required if ISTAGE = 4.

FREQEN(J) = natural frequencies.

VIII DATA CARD

* BLANK, DOT, STAR

* FORMAT(3 A 1)

This card is required if IPLOT = 1.

BLANK = blank space

DOT = .

STAR = *

APPENDIX C

```
oooooooooooooooooooooooooooooooooooooooooooooooooo
THE NON-ZERO ELEMENTS OF THE STIFFNESS MATRIX ARE:
oooooooooooooooooooooooooooooooooooooooooooooooooo
K11 = -OMEGAS * M * (L + 2.*C9*L2 + C9*C9*L3)

K22 = 4.*C5*C5*L3 + 12.*C5*C6*L4 + 9.*C6*C6*L5
TEM = L - 2.*C5*L3 - 2.*C6*L4 + C5*C5*L5 +2.*C5*C6*L6 +C6*C6*L7
K22 = C11*K22 - OMEGAS*M*TEM

K33 = C1*C1*L3 + 2.*C1*C2*L4+ C2*C2*L5
K33 = K33 * C11

K42 = L + C9*L2 + C5*L3 + (2.*C6 + C5*C9)*L4 + 2.*C6*C9*L5
TEM = C5*L*L + (3.*C6 + 2.*C5*C9)*L3 + 3.*C6*C9*L4
K42 = (K42 + XJ*TEM)*OMEGAS*M*E*SBETA

K43 = C1*L3 + (C2 + C1*C9)*L4 + C2*C9*L5
TEM = C1*L2 + (C2 + C1*C9)*L3 + C2*C9*L4
K43 = -(K43 + XJ*TEM)*OMEGAS * M * E * CBETA

K44 = L + 2.*C9*L2 + C9*C9*L3
K44 = OMEGAS * (KM2S - KM1S) * CTBETA * K44

K53 = -C1*L2 + (2*C1*C3 - C2)*L3
K53 = -(K53 + (C1*C4 + 2.*C2*C3)*L4 + C2*C4*L5)*C11

K54 = -L2 + (2.*C3 - C9)*L3 + (2.*C3*C9 + C4)*L4 + C4*C9*L5
TEM = -L + (2.*C3 - C9)*L2 + (2.*C3*C9 + C4)*L3
TEM = XJ*(TEM + C4*C9*L4)
K54 = OMEGAS * M * E * CBETA * (TEM + K54)

K55 = L-4.*C3*L2+2.*(2.*C3*C3-C4)*L3+4.*C3*C4*L4+C4*C4*L5
K55 = C11*K55

K62 = 2.*C5*L2+(3.*C6 + 8.*C5*C7)*L3 + 6.*(2.*C6*C7 + C5*C8)*L4
K62 = (K62 + 9*C6*C8*L5)*(-C11)
TEM = L2 + 2.*C7*L3 + (C8 - C5)*L4 - (C6 + 2.*C5*C7)*L5
      - (2.*C6*C7 + C5*C8)*L6 - C6*C8*L7
K62 = K62 - TEM*OMEGAS*M

K64 = 2.*C7*L3 + 2.*(C7*C9 + C8)*L4 + 2.*C8*C9*L5
TEM = (L + (C9+4.*C7)*L2 + (4.*C7*C9+3.*C8)*L3 + 3.*C8*C9*L4)*XJ
K64 = -OMEGAS*M*E*SBETA*(TEM + K64)

K66=L+8.*C7*L2+(16.*C7*C7+6.*C8)*L3+24.*C7*C8*L4+9.*C8*C8*L5
TEM = L3+4.*C7*L4+(4.*C7*C7+2.*C8)*L5+4.*C7*C8*L6+C8*C8*L7
K66 = C11 * K66 - OMEGAS*M*TEM

K71 = -OMEGAS*M*C10*(L2 + C9*L3)

K77 = -OMEGAS * M * C10 * C10 * L3

K82 = -C11*(4.*C5*C5*L3 + 12.*C5*C6*L4 + 9.*C6*C6*L5)
TEM = C5*L3 + C6*L4 - C5*C5*L5 - 2.*C5*C6*L6 - C6*C6*L7
K82 = K82 - TEM*OMEGAS*M
```

```
K84 =-OMEGAS*M*E*SBETA*(C5*L3 + (2.*C6+C5*C9)*L4 + 2.*C6*C9*L5)
TEM = XJ*(2.*C5*L2 + (3.*C6+2.*C5*C9)*L3 + 3.*C6*C9*L4)
K84 = K84 - OMEGAS*M*E*SBETA*TEM

K86 =2.*C5*L2 + (8.*C5*C7 + 3.*C6)*L3 + 6.*(C5*C8 + 2.*C6*C7)*L4
     + 9.*C6*C8*L5
TEM = C5*L4 + (2.*C5*C7 + C6)*L5+(C5*C8 + 2.*C6*C7)*L6 + C6*C8*L7
K86 = C11*K86 - OMEGAS*M*TEM

K88 = 4.*C5*C5*L3 + 12.*C5*C6*L4 + 9.*C6*C6*L5
TEM = C5*C5*L5 +2.*C5*C6*L6 + C6*C6*L7
K88 = C11 * K88 - OMEGAS * M * TEM

K93 = -C11 * (C1*C1*L3 + 2.*C1*C2*L4 + C2*C2*L5)

K94 = C1*L3 + (C1*C9 + C2)*L4 + C2*C9*L5
TEM = XJ*(C1*L2 + (C1*C9 + C2)*L3 + C2*C9*L4)
K94 = OMEGAS*M*E*CBETA*(K94 + TEM)

K95 = -C1*L2 + (2*C1*C3 - C2)*L3 + (C1*C4 + 2.*C2*C3)*L4
K95 = (K95 + C2*C4*L5)*C11

K99 = C11*(C1*C1*L3 + 2.*C1*C2*L4 + C2*C2*L5)

K102 = L2 + C5*L4 + 2.*C6*L5
TEM =   2.*C5*L3 + 3.*C6*L4
K102 = OMEGAS*M*E*SBETA*C10*(K102 + XJ * TEM)

K103 = -OMEGAS*M*E*CBETA*C10*(C1*L4+C2*L5+XJ*(C1*L3 + C2*L4))

K104 = OMEGAS*(KM2S-KM1S)*C10*CTBETA*(L2 + C9*L3)

K105 = 2.*C3*L4-L3+C4*L5 + XJ*(2.*C3*L3 - L2 + C4*L4)
K105 = OMEGAS*M*E*CBETA*C10*K105

K106 = 2.*(C7*L4  + C8*L5)
TEM =   L2 + 4.*C7*L3 + 3.*C8*L4
K106 = -OMEGAS*M*E*SBETA*C10*(K106 + XJ*TEM)

K108 = C5*L4 + 2.*C6*L5
TEM =   2.*C5*L3 + 3.*C6*L4
K108 = -OMEGAS*M*E*SBETA*C10*(K108 + XJ*TEM)

K109 = OMEGAS*M*E*CBETA*C10*(C1*L4 + C2*L5 + XJ*(C1*L3 + C2*L4))

K1010 = OMEGAS * (KM2S - KM1S) * C10 * C10 * CTBETA * L3

K113 = -C11*(C1*C3*L3 + (C1*C4 + C2*C3)*L4 + C2*C4*L5)

K114 = C3*L3 + (C4 + C3*C9)*L4 + C4*C9*L5
TEM = C3*L2 + (C4 + C3*C9)*L3 + C4*C9*L4
K114 = OMEGAS*M*E*CBETA*(K114 + XJ*TEM)

K115 = C11*(-C3*L2+(2.*C3*C3-C4)*L3+3.*C3*C4*L4+C4*C4*L5)

K119 = C11*(C1*C3*L3 + (C1*C4 + C2*C3)*L4 + C2*C4*L5)
```

```
K1110 = C3*L4 + C4*L5 + XJ*(C3*L3 + C4*L4)
K1110 = OMEGAS*M*E*CBETA*C10*K1110

K1111 = C11*(C3*C3*L3 + 2.*C3*C4*L4 + C4*C4*L5)

K122 = -C11*(4.*C5*C7*L3 + 6.*(C6*C7+C5*C8)*L4 + 9.*C6*C8*L5)
TEM = C7*L3 + C8*L4 - C5*C7*L5 - (C6*C7 + C5*C8)*L6 - C6*C8*L7
K122 = K122 - OMEGAS*M*TEM

K124 = C7*L3 + (2.*C8 + C7*C9)*L4 + 2.*C8*C9*L5
TEM = XJ*(2.*C7*L2 + (3.*C8 + 2.*C7*C9)*L3 + 3.*C8*C9*L4)
K124 = -OMEGAS*M*E*SBETA*(K124 + TEM)

K126 =2.*C7*L2+(8.*C7*C7+3.*C8)*L3+18.*C7*C8*L4+9.*C8*C8*L5
TEM = C7*L4 + (2.*C7*C7 + C8)*L5 + 3.*C7*C8*L6 + C8*C8*L7
K126 = C11*K126 - OMEGAS*M*TEM

K128 = 4.*C5*C7*L3 + 6.*(C6*C7 + C5*C8)*L4 + 9.*C6*C8*L5
TEM = C5*C7*L5 + (C6*C7 + C5*C8)*L6 + C6*C8*L7
K128 = C11*K128 - OMEGAS*M*TEM

K1210 = C7*L4 + 2.*C8*L5 + XJ*(2.*C7*L3 + 3.*C8*L4)
K1210 = -OMEGAS*M*E*SBETA*C10*K1210

K1212 = 4.*C7*C7*L3 + 12.*C7*C8*L4 + 9.*C8*C8*L5
TEM = C7*C7*L5 + 2.*C7*C8*L6+ C8*C8*L7
K1212 = C11*K1212 - OMEGAS*M*TEM
```

where

OMEGAS = Ω^2 ; C1 = $6/\ell^2$; C2 = $-6/\ell^3$; C3 = $2/\ell$;

C4 = $-3/\ell^2$; C5 = $3/\ell^2$; C6 = $-2/\ell^3$; C7 = $-1/\ell$;

C8 = $1/\ell^2$; C9 = $-1/\ell$; C10 = $1/\ell$; C11 = T ;

L2 = $\ell^2/2$; L3 = $\ell^3/3$; L4 = $\ell^4/4$; L5 = $\ell^5/5$;

L6 = $\ell^6/6$; L7 = $\ell^7/7$

ℓ = length of the element ; M = mass per unit length;

XJ = distance of the element from the axis of rotation;

E = distance between mass and elastic axes, e;

SBETA = $\sin\beta$; CBETA = $\cos\beta$

KM1S = $k_{m_1}^2$; KM2S = $k_{m_2}^2$; CTBETA = $\cos 2\beta$

$$T = \int_{\ell_e}^{R} \Omega^2 mx\, dx \quad ; \quad \ell_e = \ell/2 + XJ$$